

# **COMPUTER AIDED DESIGN OF SUBSTATION SWITCHING SCHEMES**

A Thesis

Submitted to the College of Graduate Studies and Research

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

in the

Department of Electrical Engineering

University of Saskatchewan

by

**Pramod Dhakal**

Saskatoon, Saskatchewan, Canada

October 2000

The author claims copyright. Use shall not be made of the material contained herein  
without proper acknowledgment, as indicated on the copyright page.



**National Library  
of Canada**

**Acquisitions and  
Bibliographic Services**

**395 Wellington Street  
Ottawa ON K1A 0N4  
Canada**

**Bibliothèque nationale  
du Canada**

**Acquisitions et  
services bibliographiques**

**395, rue Wellington  
Ottawa ON K1A 0N4  
Canada**

*Your file Votre référence*

*Our file Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-63860-X

**Canada**

## **COPYRIGHT**

The author has agreed that the Library, University of Saskatchewan, may make this thesis freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this thesis for scholarly purposes may be granted by the Professor who supervised the thesis work recorded herein or, in his absence, by the Head of the Department or the Dean of the College in which the thesis work was done. It is understood that due recognition will be given to the author of this thesis and to the University of Saskatchewan in any use of the material in this thesis. Copying or publication or any other use of this thesis for financial gain without approval by the University of Saskatchewan and the author's written permission is prohibited.

Requests for permission to copy or to make any other use of the material in this thesis in whole or in part should be addressed to:

Head of the Department of Electrical Engineering  
University of Saskatchewan  
Saskatoon, Canada S7N 0W0

## **ACKNOWLEDGEMENTS**

The author expresses his deep gratitude and appreciation to Prof. M.S. Sachdev and Prof. T.S. Sidhu for their supervision of this work. Proper advice at critical moments and continuous encouragements received from the professors made the author to stay focussed till the completion of this work. Opportunity provided by Prof. Sachdev to work in this project, which provided an interesting blend of experiences that proved to be valuable in the professional career of the author, is deeply appreciated. His special interest in the project have been greatly helpful.

The author expresses his appreciations and thanks to Prof. Garry Walker, Prof. Kunio Takaya, Prof. Spiro Yanacopolus, Prof. Ron Bolton, Prof. Nurul Chaudhury, who provided for valuable guidance and advice during this research. Sincere thanks go to Prof. Tony Kusalik and Prof. Jean Paul Tremblay for providing valuable time and suggestions, specially for providing good insight in object-oriented techniques and algorithm development. Special thanks to Mr. I.J. Macphedran for his help in the software setup and debugging.

The author expresses his deep appreciation to friends who helped him in many ways during his studies. Special thanks go to Heichman, Karki, Shakya, Adhikari, Tiwari, and Dyck families. Encouragements received from Dr. Anant Jalnapurkar, Mr. Rajesh Karki, Mrs. Jeannine Minielly and Mr. Joe Monahan during the last stretch of the completion of the work are greatly appreciated. Encouragements from Prof. Sachdev, Prof. Sidhu and Dr. Jalnapurkar were especially important and are highly regarded by the author. Special thanks go to Mrs. Sachdev for her wonderful support.

Sincere thanks go to author's family whose patience, support and help were important for his success. His son, Prashanta, is specially thanked for typing some of the content of the thesis, drawing of some figures and preparing the table of content.



## **ABSTRACT**

Every switching operation in a power substation must be executed after proper evaluation of its consequence. Often, switching actions for various operating conditions and contingencies are pre-determined by using the control engineer's knowledge and previous experience. When needed, these actions are carried out by either a computer or an operator. However, it is difficult for an operator to know all the operating conditions and contingencies in advance. An on line decision-making tool can provide an operator with the most appropriate switching decisions for the present system configuration and system state so the continuity and quality of power supply can be maintained by implementing those switching decisions. Since the circuit configuration varies from one substation to another, such decision making tools developed for one substation may not be applicable to another substation.

This thesis presents a technique to design digital computer based substation switching schemes. The technique is applicable to all practical substations. The rules of interlocking and sequence switching used by the schemes are generalized and are not based upon a particular substation configuration. While being applicable to all substations, these schemes also ensure the most appropriate switching operations with minimum operator intervention. The switching schemes ensure the safety of the system before making any switching action. They assist human operators in the evaluation of abnormal circumstances and are easy to integrate with other monitoring, control and protection systems.

A software application for developing switching schemes for substations was developed by using the techniques presented in this thesis. Switching schemes were developed for eight substations by using the semi-automated design tools provided with the application. The switching schemes were then tested for their correct operation. Results of the tests showed that the technique developed by the work presented in this thesis can provide appropriate switching scheme for a substation of any practical configuration.

## Table of Contents

COPYRIGHT	i
ACKNOWLEDGMENTS	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	x
LIST OF TABLES	xv
 1. INTRODUCTION	 1
1.1. Electric Power Systems	1
1.2. Role of Substations in a Power System	2
1.3. Trends in Substation Operation	3
1.4. Objectives of the Research	5
1.5. History of the Research	6
1.6. Contributions of the Research	6
1.7. Outline of the Thesis	7
 2. SUBSTATIONS	 10
2.1. Composition and Placement	10
2.2. Role of Substation Equipment	14
2.2.1. Circuit breakers	14
2.2.2. Load-break switches	14
2.2.3. Isolating switches	15
2.2.4. Grounding switches	16
2.2.5. Fuse switches	16
2.2.6. Busbars	16
2.2.7. Transformers	17
2.2.8. Reactors	17
2.2.9. Instrument transformers	17
2.3. Substation Configurations	18
2.3.1. Single busbar arrangement	18
2.3.2. Sectionalized single busbar arrangement	21
2.3.3. Double busbar arrangement	22

2.3.4.	Double busbar with by-pass isolator arrangement	23
2.3.5.	Two circuit breaker arrangement	24
2.3.6.	Sectionalized duplicate busbar arrangement	25
2.3.7.	Main-and-transfer bus arrangement	26
2.3.8	Ring busbar arrangement	27
2.3.9.	Breaker-and-a-half arrangement	28
2.3.10	Multiple bus arrangement	29
2.4.	Summary	30
3.	SUBSTATION SWITCHING	31
3.1.	Types of Switching Operations	31
3.2.	Interlocking	32
3.3.	Developments in Interlocking Techniques	34
3.4.	Sequence Switching	37
3.5.	Developments in Substation Switching	39
3.5.1.	Use of expert systems	41
3.6.	Summary	43
4.	SWITCHING DATA AND PROCEDURES	44
4.1.	Basic Switching Procedures	44
4.1.1.	Opening and closing of a switch	44
4.1.2.	Bypassing a component	45
4.1.3.	Operating switches in a branch	45
4.1.4.	Connecting a line to a busbar	45
4.1.5.	Disconnecting a line from a busbar	45
4.1.6.	Connecting busbars	46
4.2.	Advance Switching Procedures	46
4.2.1.	Identifying equipment to be taken out of service	46
4.2.2.	Determining of isolation boundary	46
4.2.3.	Identifying affected circuits	47
4.2.4.	Rerouting of flow of power	47
4.2.5.	Isolating an equipment	47
4.2.6.	Grounding an equipment	47
4.2.7.	Restoring circuits	48
4.3.	Formulation of Constraints	48
4.3.1.	General	49

4.3.2.	Circuit breakers	49
4.3.2.1.	Opening operation	49
4.3.2.2.	Closing operation	50
4.3.3.	Load-break switches	50
4.3.4.	Isolators	50
4.3.5.	Grounding switches	51
4.3.6.	Transformers	51
4.3.7.	Sequence switching	52
4.3.8.	Substation specific constraints	52
4.4.	Data Requirements and Data Structures	53
4.4.1.	Switches	53
4.4.2.	Nodes	54
4.4.3.	Busbars	55
4.4.4.	Input-output nodes	56
4.4.5.	Transformers	57
4.4.6.	Reactors and capacitors	57
4.4.7.	Measuring instruments	57
4.4.8.	Substation	60
4.4.9.	Power System	61
4.4.10.	Bus data of the system	62
4.4.11.	Line data of the system	63
4.5.	Levels of Interlocks	63
4.5.1.	Component level interlocks	63
4.5.2.	Substation level interlocks	64
4.5.3.	System level interlocks	65
4.5.4.	Manual interlocks	65
4.6.	Summary	66
5.	SWITCHING TECHNIQUE - MAJOR COMPONENTS	67
5.1.	Status Handling	67
5.1.1.	Uniform status storage and processing protocol	68
5.1.2.	Syntax for specifying status	72
5.1.3.	Handling of on-line status	73
5.1.4.	Handling of power flow results	74
5.2.	Substation Topology	75
5.2.1.	Topology representation	75

5.2.2.	Topology determination	76
5.3.	Searching of Power Flow Paths	82
5.4.	Determination of Switching Sequence	84
5.4.1.	Connecting two nodes	84
5.4.2.	Disconnecting two nodes	86
5.5.	Generalized Sequence Switching Technique	88
5.6.	Generalized Interlocking Technique	92
5.7.	Summary	94
6.	IMPLEMENTATION	95
6.1.	Tools and Techniques	95
6.2.	Substation Modelling	96
6.2.1.	Convention	96
6.2.2.	Composition of a switching system	97
6.2.3.	Component relationships	99
6.3.	Data Compilation	100
6.4.	Graphical User Interface	102
6.4.1.	Role of WIU	103
6.4.2.	Tool box and substation diagrams	104
6.5.	Editing Interlocks	105
6.6.	Implementation of Status Management Unit	106
6.7.	Implementation of Interlock Management Unit	108
6.7.1.	Implementation of Lock Units	108
6.7.2.	Compilation of text based expressions	109
6.7.3.	Evaluation of interlocks	110
6.7.4.	Determination of locked state	111
6.8.	Implementation of Command Handler Unit	111
6.8.1.	Switching operations	112
6.8.1.1.	Close or open a switch	112
6.8.1.2.	Connect a node to another node	113
6.8.1.3.	Disconnect a node from another node	113
6.8.1.4.	Isolate node	114
6.8.1.5.	Isolate a series component	114
6.8.1.6.	Bus transfer	115
6.8.2.	Supporting modules	115

6.8.2.1.	Topology tracer	115
6.8.2.2.	Energized node detector	117
6.8.2.3.	Detector of the making or breaking of currents	118
6.8.2.4.	Path tracer	119
6.9.	Integrating with System Data	119
6.9.1.	Load flow data formats	120
6.9.2.	Linking the substation data with the load-flow data	120
6.9.3.	The role of the user in the integration	121
6.10.	Graph Plotting	122
6.10.1.	Prerequisites for running the graph plotting program	122
6.11.	Summary	124
7.	RESULTS	125
7.1.	Test Substations	125
7.1.1.	Test Substation 1	125
7.1.2.	Test Substation 2	126
7.1.3.	Test Substation 3	126
7.1.4.	Test Substation 4	126
7.1.5.	Test Substation 5	126
7.1.6.	Test Substation 6	126
7.1.7.	Test Substation 7	127
7.1.8.	Test Substation 8	127
7.2.	Preparation of Switching Schemes	127
7.2.1.	Preparing of user interface	127
7.2.2.	Entering and initializing component data	128
7.2.3.	Developing interlock logic	128
7.3.	Complex Interlocking Schemes	128
7.4.	Testing Interlocking Schemes	133
7.5.	Switching Sequence Test	137
7.5.1.	Connecting and disconnecting lines	137
7.5.2.	Transferring busbars	138
7.5.3.	Isolating switches	139
7.5.4.	Isolating busbars	140
7.5.5.	Bypassing faulty equipment	141

7.6. Summary	142
8. SUMMARY CONCLUSION AND FUTURE WORK	171
8.1. Summary	171
8.2. Conclusion	174
8.3. Suggestions for Future Work	175
REFERENCES	176
APPENDICES	
A. Decoupled Power Flow Calculations	181
B. Selection of Programming Language and Development Environment	192
C. Major Classes in the SSP	193
D. Graphical User Interface	194
E. Converting Infix Expressions into Postfix Expressions	196
F. Load Flow Data Formats	198
G. Load-flow Input Data for a 9 Bus Test System	203
H. Test Results	205

## LIST OF FIGURES

<b>Figure 2.1</b>	Substations in a sample power system.	11
<b>Figure 2.2</b>	Symbols of equipment commonly used in substations.	13
<b>Figure 2.3</b>	Location of transducers in a 765kV substation.	19
<b>Figure 2.4</b>	A typical single-busbar system.	20
<b>Figure 2.5</b>	A typical sectionalized single-busbar system.	21
<b>Figure 2.6</b>	A typical double busbar arrangement.	22
<b>Figure 2.7</b>	A typical double busbar with by-pass isolator arrangement.	24
<b>Figure 2.8</b>	A typical two circuit breaker system.	25
<b>Figure 2.9</b>	A sectionalized two circuit breaker system.	26
<b>Figure 2.10</b>	A main-and-transfer busbar arrangement.	27
<b>Figure 2.11</b>	A ring busbar arrangement.	28
<b>Figure 2.12</b>	A breaker-and-a-half system.	29
<b>Figure 2.13</b>	A multiple busbar system.	30
<b>Figure 3.1</b>	A single busbar substation and its equivalent circuits.	35
<b>Figure 3.2</b>	A feeder circuit and its interlock logic circuit.	36
<b>Figure 3.3</b>	A flowchart for checking the security of a substation.	38
<b>Figure 3.4</b>	Initial and final configurations of a substation.	42
<b>Figure 4.1</b>	A substation with bus coupling circuit.	51
<b>Figure 5.1</b>	A segment of a substation circuit.	77
<b>Figure 5.2</b>	Illustration of a shunt path.	82
<b>Figure 5.3</b>	A bus-transfer process under progress.	89
<b>Figure 5.4</b>	A section of a double-bus substation.	91
<b>Figure 6.1</b>	Composition of a switching system.	98
<b>Figure 6.2</b>	Decomposition of components into specialized classes.	98
<b>Figure 6.3</b>	Relationship of a component with various modules.	99
<b>Figure 6.4</b>	The toolbox.	105
<b>Figure 6.5</b>	Sequence of events in interlock modification.	106
<b>Figure 6.6</b>	Logical relationship of DLS, LS and OLS with LOCKED status.	111
<b>Figure 7.1</b>	A section of a double busbar substation.	143
<b>Figure 7.2</b>	A section of a main-and-transfer busbar substation.	144
<b>Figure 7.3</b>	A combination of double busbar and two circuit-breaker sections.	145
<b>Figure 7.4</b>	A part of a sectionalized double busbar substation.	146



<b>Figure 7.5</b>	A substation with a combination of double busbar and main-and-transfer busbar sections.	147
<b>Figure 7.6</b>	A 765V double busbar substation with bypass isolators.	148
<b>Figure 7.7</b>	A 400kV double-busbar substation with by-bass isolators.	149
<b>Figure 7.8</b>	A three section substation.	150
<b>Figure 7.9</b>	Data being entered for a switch in the GUI environment of the SSP.	151
<b>Figure 7.10</b>	Circuit segment patterns that build Test Substation 6.	152
<b>Figure 7.11</b>	Command being issued to open isolator 11.	153
<b>Figure 7.12</b>	The SSP refuses to operate isolator 11.	154
<b>Figure 7.13</b>	Command being issued to open circuit breaker B3.	155
<b>Figure 7.14</b>	The SSP warns about the consequences and asks the operator to confirm the operation.	156
<b>Figure 7.15</b>	The SSP opens the circuit breaker B3.	157
<b>Figure 7.16</b>	Command being issued to close grounding switch G41.	158
<b>Figure 7.17</b>	The SSP refuses to close G41 and displays the interlock constraint that blocked the operation.	159
<b>Figure 7.18</b>	Command being issued to connect line 7L4 to a busbar.	160
<b>Figure 7.19</b>	The status of switches after the SSP connects line 7L4 to Bus1.	161
<b>Figure 7.20</b>	Command being issued to transfer lines from Bus1 to Bus2.	162
<b>Figure 7.21</b>	The substation after the SSP transferred lines from Bus1 to Bus2.	163
<b>Figure 7.22</b>	Command is being issued to isolate circuit breaker C4.	164
<b>Figure 7.23</b>	The substation after the SSP isolated circuit breaker C4.	165
<b>Figure 7.24</b>	Command being issued to isolate Bus1 on Test Substation 1.	166
<b>Figure 7.25</b>	The state of the substation after isolating Bus1.	167
<b>Figure 7.26</b>	Command being issued to bypass circuit breaker C4 from power flow paths to be formed in future.	168
<b>Figure 7.27</b>	Command being issued to connect line 7L4 to the busbar after issuing the command to bypass C4.	169
<b>Figure 7.28</b>	Connection made between line 7L4 and Bus1 while bypassing C4.	170
<b>Figure A.1</b>	Procedure for load flow calculations.	189
<b>Figure A.2</b>	Procedure for LU factorization of Jacobian matrix.	190
<b>Figure G.1</b>	Power flow input data of the nine bus power system.	203
<b>Figure G.1</b>	Single line diagram of the nine bus power system..	204
<b>Figure H.1</b>	Command being issued to close isolator 4.	222

<b>Figure H.2</b>	The SSP refuses to close isolator 4.	223
<b>Figure H.3</b>	Command being issued to open isolator 4 after the bus coupler circuit was closed.	224
<b>Figure H.4</b>	The SSP closes isolator 4.	225
<b>Figure H.5</b>	Command being issued to open isolator 2.	226
<b>Figure H.6</b>	The SSP refuses to open isolator 2.	227
<b>Figure H.7</b>	Command being issued to open isolator 4.	228
<b>Figure H.8</b>	The SSP opens isolator 4.	229
<b>Figure H.9</b>	Command being issued to open isolator 13.	230
<b>Figure H.10</b>	The SSP opens isolator 13.	231
<b>Figure H.11</b>	Command being issued to close isolator 8.	232
<b>Figure H.12</b>	The SSP refuses to close isolator 8.	233
<b>Figure H.13</b>	Command being issued to open isolator 13.	234
<b>Figure H.14</b>	The SSP refuses to open isolator 13.	235
<b>Figure H.15</b>	Command being issued to open circuit breaker B2.	236
<b>Figure H.16</b>	The SSP warns about the interruption of current and gibes the operator a chance to cancel the operation.	237
<b>Figure H.17</b>	Command being issued to close isolator 1.	238
<b>Figure H.18</b>	The SSP has closed isolator 1 and a new command to close isolator 2 is being issued.	239
<b>Figure H.19</b>	The SSP has closed isolator 2 and a command is being issued to close circuit breaker B5.	240
<b>Figure H.20</b>	The SSP has closed circuit breaker B5 and a command is being issued to close isolator 6.	241
<b>Figure H.21</b>	The SSP has closed isolator 6 and a command is being issued to open circuit breaker B2.	242
<b>Figure H.22</b>	The SSP has opened circuit breaker B2 and a command is being issued to open isolator 5.	243
<b>Figure H.23</b>	Command being issued to open isolator 12.	244
<b>Figure H.24</b>	The SSP opens isolator 12.	245
<b>Figure H.25</b>	Command being issued to open isolator I43.	246
<b>Figure H.26</b>	The SSP refuses to operate I43 and displays the reason for the refusal.	247
<b>Figure H.27</b>	Command being issued to open circuit breaker C4.	248
<b>Figure H.28</b>	The SSP warns about the interruption of current, and gives a chance to cancel the closing operation.	249
<b>Figure H.29</b>	The SSP has closed C4 and a command is being issued to open isolator I43.	250

<b>Figure H.30</b>	The SSP has opened isolator I43 and a command is being issued to open isolator I41.	251
<b>Figure H.31</b>	The SSP has opened I41 and enabled grounding switches G41 and G42.	252
<b>Figure H.32</b>	Command being issued to close grounding switch G42.	253
<b>Figure H.33</b>	The SSP closes grounding switch G42 and disables isolator I43 and circuit breaker C4.	254
<b>Figure H.34</b>	Command is being issued to close grounding switch G41.	255
<b>Figure H.35</b>	The SSP has closed grounding switch G41 and has disabled isolators I41 and I42.	256
<b>Figure H.36</b>	Command is being issued to close circuit breaker C0.	257
<b>Figure H.37</b>	Switches I12, I22, I32, I43 and I52 are enabled.	258
<b>Figure H.38</b>	Command being issued to disconnect Load1 from the bus.	259
<b>Figure H.39</b>	The SSP disconnects Load1 from Bus1.	260
<b>Figure H.40</b>	Command being issued to disconnect Source1 from busbars.	261
<b>Figure H.41</b>	The SSP disconnects Source 1 from busbars.	262
<b>Figure H.42</b>	Command being issued to connect Source1 to Load1.	263
<b>Figure H.43</b>	The SSP connects Source1 with Load1.	264
<b>Figure H.44</b>	Command to disconnect Load1 from the bus.	265
<b>Figure H.45</b>	State of the substation after disconnecting Load1 and Load3 from the busbars.	266
<b>Figure H.46</b>	Command to connect Load1 to the busbar.	267
<b>Figure H.47</b>	Load1 and Load3 connect back to the busbars.	268
<b>Figure H.48</b>	Declaring isolator 9 as unhealthy.	269
<b>Figure H.49</b>	Connecting Load2 to the busbar.	270
<b>Figure H.50</b>	Load2 is connected to Bus1 while avoiding switch 9.	271
<b>Figure H.51</b>	Command being issued to discount line 7L4 from the busbars.	272
<b>Figure H.52</b>	The status of switches after the SSP disconnects line 7L4 from the busbars.	273
<b>Figure H.53</b>	Command being issued to transfer lines from Bus1 to Bus2.	274
<b>Figure H.54</b>	The substation after the SSP transfers the lines from Bus1 to Bus2.	275
<b>Figure H.55</b>	Command being issued to isolate circuit breaker B6.	276
<b>Figure H.56</b>	The substation after the isolation of B6 and the switching sequence required to isolate it.	277
<b>Figure H.57</b>	Command being issued to isolate Bus1 on Test Substation 5.	278
<b>Figure H.58</b>	The state of the substation after isolating Bus1.	279

<b>Figure H.59</b>	Command being issued to isolate Bus1.	280
<b>Figure H.60</b>	The status of the substation after isolating Bus1.	281
<b>Figure H.61</b>	Command being issued to declare isolator 9 as not healthy.	282
<b>Figure H.62</b>	The SSP has declared and a command is being issued to connect Load2 to the busbar.	283
<b>Figure H.63</b>	The substation after the SSP connects Load2 to Bus1 while avoiding isolator 9.	284
<b>Figure H.64</b>	Command being issued to disconnect source 7L4 from the busbar.	285
<b>Figure H.65</b>	Switching and final configuration of the substation after disconnecting line 7L4.	286
<b>Figure H.66</b>	Command being issued to bypass circuit breaker C4 from power flow paths to be formed in future.	287
<b>Figure H.67</b>	Command being issued to connect line 7L4 to the busbar after issuing the command to bypass C4.	288
<b>Figure H.68</b>	Connection made between line 7L4 and Bus1 while bypassing C4.	289
<b>Figure H.69</b>	Nine bus power system whose Bus3 represents the substation under test.	290
<b>Figure H.70</b>	Command being issued to edit data that relate the substation with the poser system.	291
<b>Figure H.71</b>	The system bus number that corresponding to the substation being specified.	392
<b>Figure H.72</b>	Load and Generator data being specified.	393
<b>Figure H.73</b>	Properties of node connecting to the rest of the power system being specified.	394
<b>Figure H.74</b>	Load-flow program being launched.	395
<b>Figure H.75</b>	Graph plotting program being launched.	396
<b>Figure H.76</b>	Conclusion of the load-flow analysis.	397
<b>Figure H.77</b>	Command being selected to plot and the actual plot of the voltage levels at system buses.	398
<b>Figure H.78</b>	A plot of the current magnitudes at system lines.	399
<b>Figure H.79</b>	Display of the load-flow output data in numeric form along with the graphical plot.	300

## List of Tables

<b>Table 4.1</b>	Switching device data	54
<b>Table 4.2</b>	Node data	55
<b>Table 4.3</b>	Busbar data	56
<b>Table 4.4</b>	Input-output node data	58
<b>Table 4.5</b>	Transformer data	59
<b>Table 4.6</b>	Reactor data	60
<b>Table 4.7</b>	Measuring instrument data	61
<b>Table 4.8</b>	Substation data	62
<b>Table 4.9</b>	Power system data	63
<b>Table 4.10</b>	System bus data	64
<b>Table 4.11</b>	System line data	65
<b>Table 5.1</b>	The status key words and their meanings	69
<b>Table 5.2</b>	Data before energizing Bus 1	78
<b>Table 5.3</b>	Data after energizing Bus 1	79
<b>Table 5.4</b>	Data before energizing node 7	80
<b>Table 5.5</b>	Data after energizing node 7	80
<b>Table 5.6</b>	Data before energizing node 4	81
<b>Table 5.7</b>	List of adjoining nodes	83
<b>Table 5.8</b>	Modified list of adjoining nodes	84
<b>Table 5.9</b>	Command names	90
<b>Table 6.1</b>	Structure of a class	96
<b>Table 6.2</b>	Evaluation of a postfix expression	109
<b>Table E.1</b>	Input-precedence, stack-precedence and rank values	197

## List of Acronyms

<b>SSP</b>	<b>Substation Switching Scheme Development Package</b>
<b>SMU</b>	<b>Status Management Unit</b>
<b>IMU</b>	<b>Interlock Management Unit</b>
<b>CHU</b>	<b>Command Handler Unit</b>
<b>WIU</b>	<b>Windows Interface Unit</b>
<b>WMIU</b>	<b>Windows Main Interface Unit</b>
<b>GUI</b>	<b>Graphical User Interface</b>
<b>LU</b>	<b>Lock Unit</b>
<b>LS</b>	<b>Lock String</b>
<b>DLS</b>	<b>Default Lock String</b>
<b>OLS</b>	<b>Override Lock String</b>
<b>CLOS</b>	<b>A computer language, Common Lisp Object System</b>
<b>GND</b>	<b>Grounded</b>
<b>MVA</b>	<b>Mega Volt Ampere</b>
<b>MFC</b>	<b>Microsoft Foundation Classes</b>
<b>sts</b>	<b>Status of a component</b>
<b>tssts</b>	<b>Status of a component to be tested for some state information</b>
<b>Topo</b>	<b>Topology</b>

**List of Symbols**

<b>P</b>	Active power
<b>Q</b>	Reactive power
<b>G</b>	Real part of an element of the bus-admittance matrix
<b>B</b>	Imaginary part of an element of the bus-admittance matrix
<b><math>\delta</math></b>	Phase angle of bus voltage relative to that of the slack bus
<b>R</b>	Resistance
<b>X</b>	Reactance
<b>Z</b>	Impedance
<b>J</b>	Jacobian matrix
<b>S</b>	Apparent power
<b>N</b>	Number of buses
<b>V</b>	Voltage
<b>I</b>	Current

*To my mother Dhanakala and father Dharmagat*

**- Pramod**



# **1. INTRODUCTION**

Electric power systems generate, transmit, and distribute electric energy. Switching stations are strategically placed in power systems to supply power to all customers. Switching operations are performed to direct the energy to desired routes. An improper operation can adversely affect a large part of the system. Comprehensive interlocks and computer based decision-making tools, incorporated in substation control systems, can assist operators in making appropriate switching decisions. This chapter emphasises the need for these tools and outlines the objectives for the research work that is described in this thesis.

## **1.1. Electric Power Systems**

A power system is composed of generation, transmission, and distribution subsystems. Electric power is produced in generating plants that are usually located far away from load centres. Power generated by the plants is transported over transmission lines to load centres that are scattered over wide geographical areas. The transmission lines usually form interconnected networks.

Power systems use electrical equipment such as generators, transformers, circuit breakers, reactors, capacitors and other devices. Generators convert mechanical energy to the electrical form at voltages that are mostly in the 15 kV range. Much higher voltages are used for transmitting energy over long distances so that the line losses are limited to a few percent of the transmitted energy. Conversion to higher voltages is achieved by using step-

up transformers. At the load ends, transformers are used to lower the voltage levels to 5-34 kV range. Energy is then transported to customers' locations where the voltage is further reduced for their use.

Providing power to customers at acceptable levels of quality and availability is a complex process. The performance of the system is continuously monitored, and the continuity of power supply is maintained by sometimes by reconfiguring the network as the state of the system changes. Switching and control actions are carried out in response to randomly occurring events such as equipment overloading, equipment failure, and system disturbances. When an component is isolated for maintenance, the system is reconfigured so that power supply to the customers is not interrupted. These switching and control actions are performed at the substations.

## **1.2. Role of Substations in a Power System**

Substations interconnect generation, transmission and distribution subsystems. Switching and control actions are performed at substations to maintain the continuity of power supply and to ensure that voltages at the loads are maintained at the desired levels. The continuity of power supply is maintained by reconfiguring the transmission and distribution networks as the system experiences faults, overload of equipment and voltage-limit violations caused by insufficient or excessive reactive power generation.

Changes in the configuration of a transmission network are implemented by operating appropriate switches that are usually located in substations. Switching actions taken in one substation redistribute power flows and change the voltage profile in the system. Because an improper operation can adversely effect a large part of the network, switching operations are planned and their consequences are analysed before implementing them. The purpose of the analysis is to ensure that the operations do not result in undesirable outcomes, such as equipment overload, over-voltages, under-voltages and faults.

Significant progress in the monitoring and control techniques has been made in recent decades. A brief description of these developments is provided in the next section.

### **1.3. Trends in Substation Operation**

The development of devices for making and breaking electric circuits took a long period of time. In the early days, the arc that formed on the opening of switches used to cause substantial damage to the substation equipment. Attendants had to chop the arc by using insulated hatchets. The quenching of an arc was, therefore, a major concern in substation performance. The quick-break knife-switch was perhaps the earliest form of a circuit-breaker [1].

During the first quarter of the 20th century, a more effective approach for the making and breaking of current in power circuits was developed; this consisted of immersing the knife-switches in mineral oil. The mechanism of arc formation was better understood by the late 1902's. This led to the development of arc-controlling devices. The study of arc formation and arc quenching mechanisms eventually led to the development of modern circuit-breakers.

Along with the development of circuit breakers, advancements were made in the monitoring and control systems. Protective relays detect defective lines and equipment, and abnormal operating conditions. They initiate, or permit, switching of circuit breakers. Monitoring systems continuously supervise the conditions of a power system and keep the operators informed about the state of the system. Regulating systems adjust the settings of the equipment when a parameter deviates from its predetermined limits. Co-ordinated efforts of monitoring, control and protection systems, and the system operators, make it possible to maintain continuity and quality of power supply. To make the control and protection system more versatile and comprehensive, they are now designed as microprocessors-based systems with specifically developed software programs.

Increasing demand for electricity throughout this century has forced power systems to become larger and more complex. Lines of high and extra-high voltages are being used to transmit large amounts of power. Complex switching arrangements are being used to attain the desired levels of flexibility and reliability, and to maximize the utilization of investment. With a trend towards deregulation and increased competition, utilities are loading the systems closer to their designed limits. These factors have increased the complexity in planning, design and operation of power systems. Large amounts of information must, therefore, be analysed when making switching decisions so that the integrity of the system is maintained.

Electric utilities have been using computers for planning, design, and operation of power systems since 1929 [2]. Early applications were limited due to the inadequate capacity of the computers. Power flow programs were the first large scale applications that were implemented on modern digital computers. Many new applications have been developed over the years. Analysis of system performance and effectiveness of alternative switching plans are two examples of such applications.

Despite all the advancements in digital computing technology, most operations are performed manually by operators using predefined guidelines. The exceptions are the protection and restoration switching that is done by dedicated relays. It, however, takes substantial time and tedious work to manually generate proper switching sequences. Computer-based substation switching schemes can expedite the decision making process and help avoid human errors.

Computer-based switching systems can continuously monitor the state of a power system, perform on-line analysis of the system and initiate operations to change the system configuration for maintaining its integrity. The switching decisions made in individual substations can, therefore, give proper considerations to the security of the power system.

The work that is reported in this thesis was carried out to further the development of computer based switching schemes for substations.

#### **1.4. Objectives of the Research**

Switching scheme for each substation is designed specifically for its unique configuration. Substation data, substation configurations and specifications of the switches are provided to third party contractors for developing switching schemes. Utilities incur these expenses every time the scheme is to be modified to match the changes in the configuration of a substation. The utility personnel, who have the most knowledge and experience in operating their substations, cannot develop the switching schemes on their own because proper tools for developing these schemes are not available.

The work on substation switching presently reported in the literature has focussed mainly on the development of approaches to handle interlock requirements and to generate optimal switching sequences. Procedures for implementing these approaches are complex and require knowledge of hardware and software engineering. There is no work reported in the literature for developing a framework that simplifies the procedures of developing the switching schemes.

A desirable scenario is that the engineers and programmers with specialized knowledge study the principles that govern substation switching and develop a framework that would allow operating engineers to develop switching schemes for their substations. The framework should seamlessly overcome the hardware and software engineering complexities to allow the developers to focus on the working of the substations. The motivation of this research was to develop such a framework. Keeping this in mind, the following objectives for the research work reported in this thesis were established.

1. Design a framework that accelerates the development of switching schemes without the need for considering the complexities for hardware and software implementations.

2. Automatically incorporate accepted operating practices in the switching schemes generated by the framework.
3. Examine the feasibility of developing a computer aided design and development tool based on the established framework.

#### **1.4. History of the Research**

The author as a part of his Master's thesis [3] did preliminary work for this research. Substation equipment and most commonly used substation configurations were studied. A set of generalized constraints for operating substations were established. It was perceived during the research that it is possible to develop a generalized substation-switching scheme applicable to all practical substations. Also a preliminary generalized switching scheme was developed. The approach used at the time was based on complex data structures and tightly coupled subroutines. It was realized that it is not easy to develop a robust system based on that approach.

This research project was then launched to find a robust approach. It was realized early enough that it is difficult to develop an optimal switching scheme that is also generic and takes care of all existing and future substation configurations. If attempted, this would introduce undesirable complexities in development, validation, and maintenance procedures. The development of a framework that implements only the principles governing substation switching practices was identified to be a better approach. The framework should include a mechanism for adding new constraints and syntactic and semantic rules for providing the constraints. If developed, the framework would be suitable for developing switching schemes for substations of different configurations.

#### **1.5. Contributions of the Research**

This work has made tangible contributions towards the development of substation switching schemes. Major contributions of the work include the following:

1. Procedures for analysing substation-switching problems are established.

2. A framework that has the following characteristics has been developed.
- Automatically incorporates the established practices in substation operation.
  - Provides facility, and syntactic and semantic rules, for adding new constraints for handling situations that arise in individual substations.
  - Automatically generates the scheme based on the constraints provided to the framework.
  - Provides an integrated design, development and test environment.
  - Seamlessly overcomes the complexities in developing switching schemes.
  - Facilitates an iterative development procedure so that the design is not required to be perfect the first attempt.
  - Provides a tool for system planning and system operating personnel.

The work that is reported in this thesis would allow the utilities to design switching schemes for their substations without the help of third party experts. The utilities could refine the schemes over time and modify them in future to incorporate the changes in the configurations of substations.

The seamless handling of the complexities of the schemes, automatic incorporation of accepted operating practices, and incorporating the operating practices in this framework-based approach substantially reduce the effort and time required to develop switching schemes compared to the effort and time spent on their development at this time.

## **1.6. Outline of the Thesis**

This thesis describes a technique for developing a framework that simplifies and expedites the design and implementation of safety interlocks, and generates and validates substation-switching sequences. The thesis starts by providing the background information in

substations layouts and switching operations. The technique for developing the framework, implementation of the technique, results of tests conducted on the implemented framework, and conclusions are then presented.

Substations and their role in a power system are described in Chapter 2. Major switching devices are introduced and some commonly used substation configurations are described. The switching operations are categorized considering their roles.

Different types of switching operations are described in Chapter 3. The importance of using computer-based interlocks over traditionally used mechanical interlocks is outlined. The need to operate switches in a desired sequence and issues involved in determining the sequence are presented. Chapter 3 also reviews the ongoing research in substation switching and summarises the previously published reports.

The operating constraints of switches are largely due to their design and the way the switches are used. Chapter 4 examines the operating constraints of the equipment used in substations. The data required for developing interlocking schemes and switching sequences are identified. The steps involved in determining switching sequences are identified.

The techniques for the development of generalized interlocking schemes and switching sequences are described in Chapter 5. The network graph processing technique used in this work is also presented.

The developed techniques were implemented in the form of a software package, called Substation Switching Scheme Development Package (SSP). Appropriate tools and techniques were investigated and used to implement the concepts described in Chapters 4 and 5. Object-oriented methodology was used to implement the concepts. Chapter 6 introduces the object oriented approach from the perspective of designing substation



switching schemes. The chapter also describes the techniques used to represent a substation and to implement the interlocking and sequence switching concepts.

The concepts were tested using eight substations. An analysis of the substations and sample results obtained from the tests are presented in Chapter 7. The results show that the developed concepts are valid for implementing proper interlocks in the substations, and for generating switching sequences.

A summary of the thesis and conclusions drawn from the research work are presented in Chapter 8. Suggestions for further research are also provided.

This thesis also includes seven appendices. The decoupled Newton-Raphson load flow calculation technique used in the load flow program for predicting the voltage and current profiles in a power system is presented in Appendix A. The main criteria used to select the language and tools for developing SSP are listed in Appendix B. The main classes of components used in SSP are listed in Appendix C.

Main features of the graphical user interface environment of SSP are described in Appendix D. A method of converting logical expressions from infix form to postfix form is presented in Appendix E. Two data formats, the “IEEE Common Data Format” and the “Reduced Data Format,” used to share and exchange load flow data among applications are described in Appendix F. The input data for conducting the power flow study of a sample power system are provided in Appendix G. Additional test results are presented in Appendix H to supplement those provided in Chapter 7.

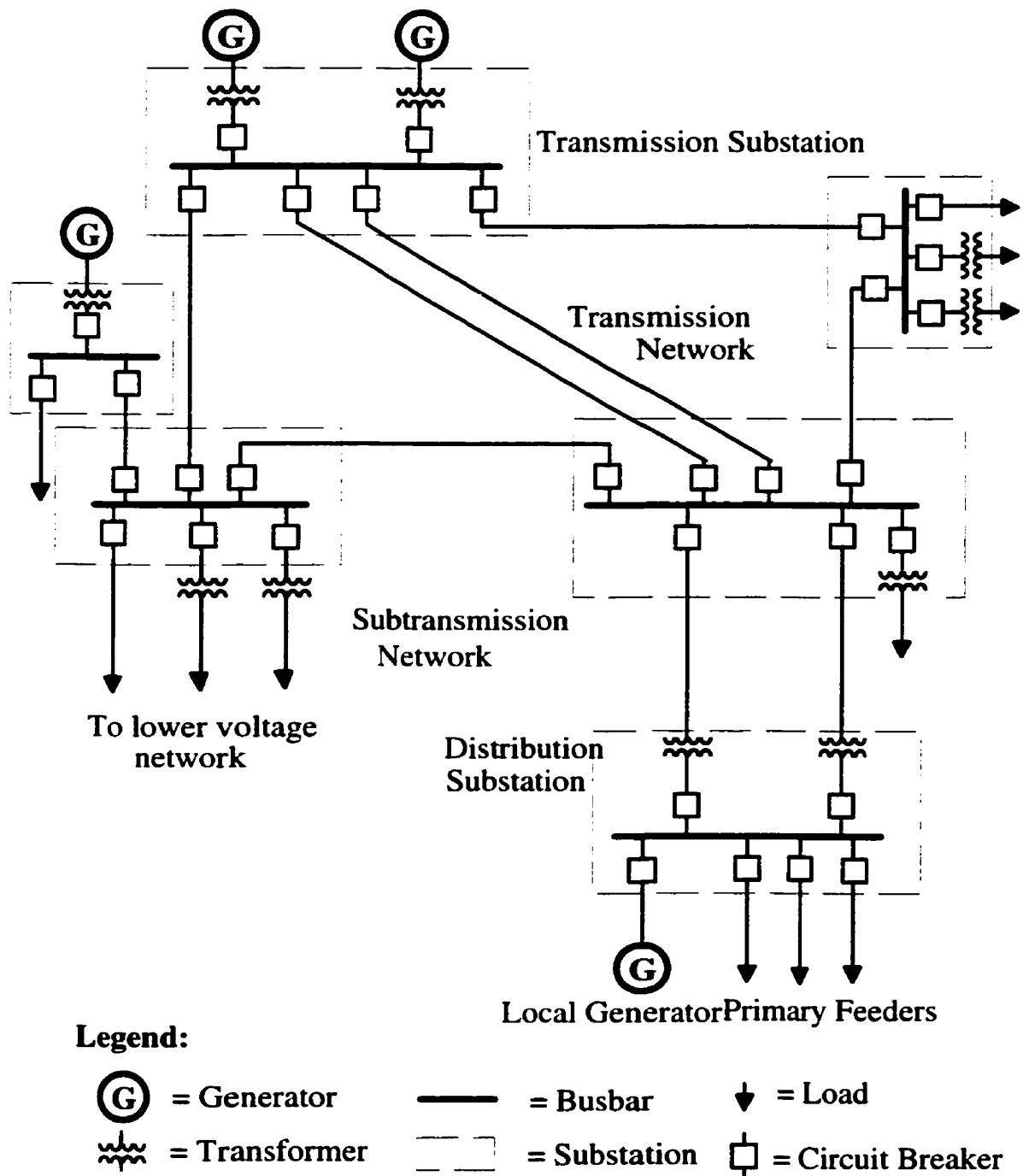
## **2. SUBSTATIONS**

Substations interface generation, transmission and distribution subsystems of a power system. Isolating switches and circuit breakers, installed in substations, perform switching operations to control power flows. Large substations are often formed by combining bays that use definite patterns for connecting input-output lines to busbars. The strength and weaknesses of a substation can be analysed by examining the topology of its bays. The concepts used in designing substations, including the layout of the equipment, are introduced in this chapter. The commonly used configurations of substations, and their advantages and disadvantages are also described.

### **2.1. Composition and Placement**

Substations are nodes of a power system network where transmission and distribution lines meet. They are composed of elements, such as circuit breakers, isolators, transformers, busbars, reactors, capacitors and grounding switches, as well as systems that protect, monitor and control them. The switches, which are provided in substations, are operated to change the topology of the network. The switches must be operated in specified orders to interrupt, restore or regulate the flow of power on lines connected through the substation.

Power lines emanating from different geographical locations converge at busbars of a substation. The voltages of the lines are changed by using transformers, so that all connections to a busbar are at a common operating voltage. Circuit breakers are placed between the lines and the busbars so that a line can be disconnected from the system when a fault occurs on it. Figure 2.1 shows a power system that has six substations and lines connecting them. Energy sources, loads, transformers, busbars and circuit breakers are shown in this figure. Isolators and ground switches, and monitoring, control and protection equipment are not shown in this figure.



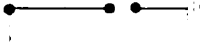
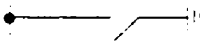
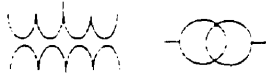



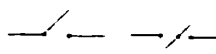
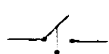
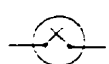

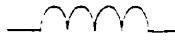
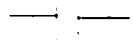
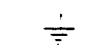
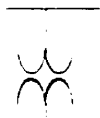
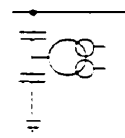
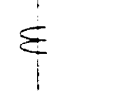
**Figure 2.1.** Substations in a sample power system.

At the substations of major generating stations, voltage is stepped up for transmitting power over long distances. These substations are called high-voltage (HV) substations, whose typical operating voltage ranges from 200 to 765 kV. A sub-transmission network connects HV substations to distribution substations that are located near load centres. The operating voltage of sub-transmission systems ranges from 69 to 150 kV. Distribution substations receive power from sub-transmission networks, transform the voltage to a lower level and deliver power to distribution feeders at voltages ranging from 2.5 to 35 kV. The distribution lines feed transformers that reduce the voltage further and supply power to consumers. Transmission and distribution substations are shown in Figure 2.1.

The switching arrangements shown in Figure 2.1 are simple. The arrangements are usually more complex than those shown in the figure. The complexity depends on several factors, such as the degree of flexibility required, the importance of the substation, and the cost of equipment.

The higher the operating voltage, the more is the damage due to faults and undesirable operations. Also, faults in HV substations can interrupt power to large areas. Transmission substations, therefore, use redundant switches and, consequently, their configurations are more complex than those of distribution substations.

Most substations have a control room and a switchyard. The monitoring, control and protection devices are housed in the control room. The control room equipment also include low voltage switches, indicating and measuring instruments, relays, control panels, communication devices and the operators' console. High voltage equipment and transducers are located in the switchyard. The high voltage equipment include circuit breakers, isolators, grounding switches, fuse switches, busbars, power transformers, lightning arresters, reactors, capacitors, current transformers, potential transformers, power-line-carrier transceivers, and other devices. Figure 2.2 shows symbols for the equipment commonly used in substations. Each equipment plays a specific role in a substation. These roles are described in the next section.

	Lightning arrester
	Grounding switch
	Transformer
	Three-winding transformer
	Bushing
	Circuit breaker
	Isolator
	Motorized isolator
	Load-break switch
	Busbar
	Reactor
	Capacitor
	Ground node
	Voltage transformer
	Capacitor-coupled voltage transformer
	Current transformer

**Figure 2.2.** Symbols of equipment commonly used in substations.

## **2.2. Role of Substation Equipment**

An equipment used in a substation plays an important role in the operation of the power system. The capabilities and limitations of the power equipment and their role in substation operation are outlined in this section.

### **2.2.1. Circuit breakers**

Circuit-breakers are used to make or break electric currents in circuits during normal operation of the system, during system faults and during system disturbances. Most high-voltage circuit breakers are physically located in the switchyard but are operated from panels provided in the control room.

A circuit-breaker is expected to have the following characteristics [1].

- i. It must be capable of closing on to and carrying full load currents.
- ii. It must have an appropriate mechanism to automatically disconnect the load under prescribed conditions.
- iii. It must be able to successfully interrupt short-circuit currents flowing through the lines controlled by it.
- iv. The gaps between its contacts must not flash-over when the circuit breaker is open.
- v. The circuit breaker, when closed on to a circuit in which a fault exists, must be able to reopen to isolate the faulted section without being damaged.
- vi. It must be capable of withstanding the flow of short-circuit currents until they are interrupted by an adjoining circuit breaker.
- vii. It must be capable of withstanding the electro-magnetic forces and thermal stresses caused by the flow of short-circuit currents.

### **2.2.2 Load-break switches**

Load-break switches, also called load-breaking fault-making switches, are used to open circuits carrying full load currents. These switches can not interrupt fault currents. Load break switches are sometimes used instead of circuit breakers. The purpose is to increase the flexibility while reducing the number of circuit breakers and, therefore, the cost of the substation.

The cost of a load-break switch is much less compared to the cost of a circuit breaker. By using a combination of a load-break switch and a fuse, fault clearing and circuit isolation can be achieved. Because of its low cost, this combination is frequently used in distribution systems.

A load-break switch must be able to perform the following functions [1]:

- i. It must be capable of breaking the rated load currents.
- ii. It must be capable of interrupting small inductive and capacitive currents experienced while disconnecting unloaded transformers, cables and overhead lines.
- iii. It must be able to close on to a faulted circuit without being damaged.
- iv. It must be capable of carrying the fault currents while they are being interrupted by another device.

### **2.2.3 Isolating switches**

Isolating switches (also called isolators, disconnects, disconnecting switches) are used to disconnect circuit-breakers, sections of busbars and parts of the system. These switches are also used to transfer circuits from one busbar to another and to provide flexibility during system operation.

An isolator is used to open a circuit only after the flow of current has been interrupted by another device. Isolators are able to carry normal as well as short circuit currents. These switches are slow moving devices but are inexpensive compared to load switches and circuit breakers.

These switches must be able to:

- i. carry normal load currents continuously,
- ii. carry fault currents until they are cleared by an interrupting device and,
- iii. make and break small currents when the voltage difference across their terminals is not significant.

The open and closed status of an isolator can be visually verified. Often, video cameras are placed at strategic locations in a switchyard to visually inspect the operation of an isolator

without leaving the control room. These cameras are remotely controlled and can be pointed towards selected equipment in the substation.

#### **2.2.4. Grounding switches**

A grounding switch is used for connecting equipment to ground after it has been isolated from the system. Before performing maintenance, an equipment must be isolated and the charge trapped on it must be drained. Impedances are often used in the ground leads for limiting the flow of currents in case the equipment happens to be energized.

Some load switches, such as oil switches used in ring main distribution substations, have integral facilities for grounding the circuit [1]. In some designs, a common switch blade moves from the "on" position to the "off" position and then to the "earth" position. In other designs, separate blades are used for line switching and for grounding. They are interlocked to prevent inadvertent grounding of the circuit while the circuit breaker is closed. The grounding must always be a deliberate action.

#### **2.2.5. Fuse switches**

These are either load switches or isolators equipped with fuse elements. The fuses are used to protect equipment and lines from over-current. Fuse switches are used instead of circuit-breakers in distribution networks. When a fuse operates to open the line, it must be manually replaced by a new one.

#### **2.2.6 Busbars**

Busbars, or buses, are conductors, or group of conductors, that serve as a common connection for two or more circuits. All power lines converging to a substation ultimately meet each other at the busbars. Electrically, they are nodes where several network branches converge.

The busbars experience forces when currents flow in them. These forces can be great when short-circuit currents flow. A busbar must be able to withstand the forces caused by the flow of fault currents.



### **2.2.7. Transformers**

A transformer is a major component of a substation. Two circuits operating at different voltages can be connected by an appropriate transformer. A power transformer can be of a two-winding, or multi-winding type. Transformers are equipped with taps for regulating the voltage at one of its terminals.

### **2.2.8. Reactors and capacitors**

Reactors and capacitors are used to regulate voltage in a power system [2]. These devices can be connected in series with a circuit or between the phase conductors and the ground. In modern substations, reactors are often thyristor controlled devices, which are called static VAR compensators.

### **2.2.9. Instrument transformers**

Crucial information on the state of the power system and the substation equipment is acquired by measuring various parameters. The measuring, monitoring, control and protection devices measure and use parameters such as current, voltage, power factor, frequency, active power, reactive power, direction of power flow, load balance, and phase angles. These parameters are measured by using two types of analog sensors, current transformers (CT) and voltage transformers (VT). These transducers provide the instantaneous values of currents and voltages. The remaining parameters are derived from these measurements. Figure 2.3 shows the placement of instrument transformers in a typical substation.

Older substations have instruments that are operated by analog signals from the transducers. In modern substations, currents and voltages are acquired in the form of quantized samples using analog to digital converters. The samples are then processed by digital signal processors to estimate the desired parameters.

## **2.3. Substation Configurations**

The following factors influence the selection of a switching arrangement for a substation:

- Locally prevailing environmental conditions
- Geographical conditions
- Importance of the substation
- Level of required operational flexibility
- Cost
- Reliability

The main goal in any design is to provide a simple arrangement that would provide the desired flexibility in operation. Analytical studies can be performed to compare the costs and benefits of different arrangements.

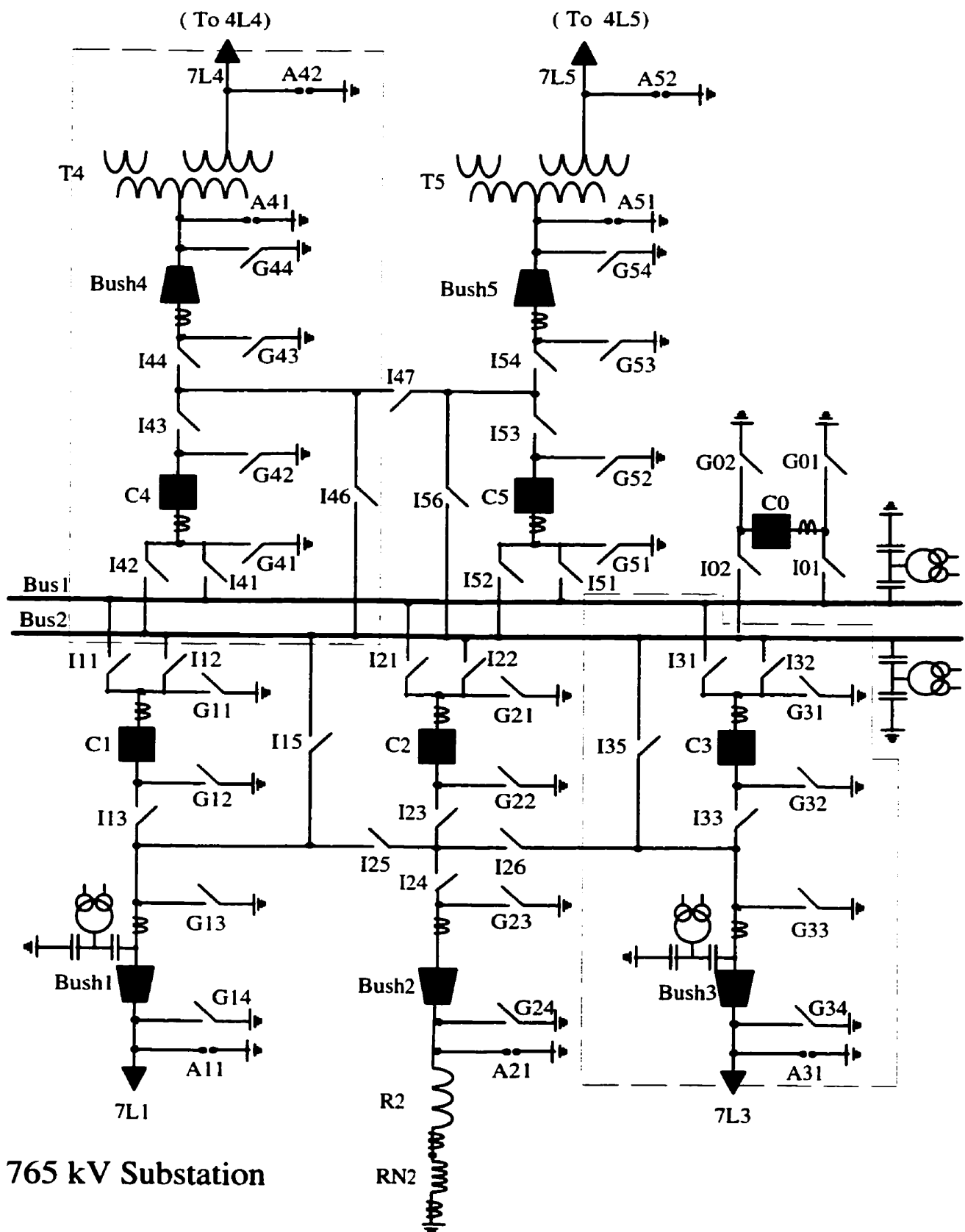
A study of the existing substation configurations shows that certain switching arrangements are often repeated. One or two typical arrangements of bays appear to be used repeatedly in a substation. For example, the substation shown in Figure 2.3 has two types of bays identified by dashed boxes.

A substation is identified by the pattern of bays used to make it. Some commonly used configurations are introduced and their advantages and disadvantages are described in this section.

### **2.3.1. Single busbar arrangement**

This is the simplest busbar arrangement in which a single set of busbars is used in the substation. All circuits are connected to this busbar. This arrangement is used in small substations and some generating stations. Figure 2.4 shows the switching arrangement of a typical single busbar substation.

A circuit breaker is placed in each line so that it can be disconnected from the rest of the system. Isolators are provided at both ends of a circuit breaker. When taking a circuit breaker, say B1 shown in Figure 2.4, out of service, the following operations are performed:



## 765 kV Substation

Figure 2.3: Location of transducers in a 765kV substation.

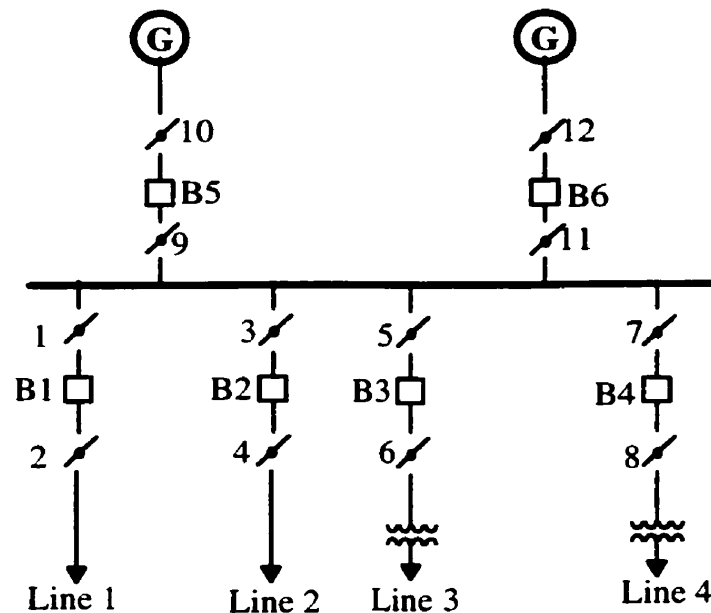
- open circuit breaker B1,
- open isolators 1 and 2.

The circuit breaker is opened to break the current flowing in the circuit. The isolators are then opened and break low levels of currents associated with the stray capacitances of the circuit.

To energize Line 1, the following operations are performed:

- close isolators 1 and 2,
- close circuit breaker B1.

The isolators are closed but they do not establish the flow of current. The circuit breaker is operated, after the contacts of the isolators have reached their final positions, to establish the flow of current in the circuit.



**Figure 2.4.** A typical single-busbar system

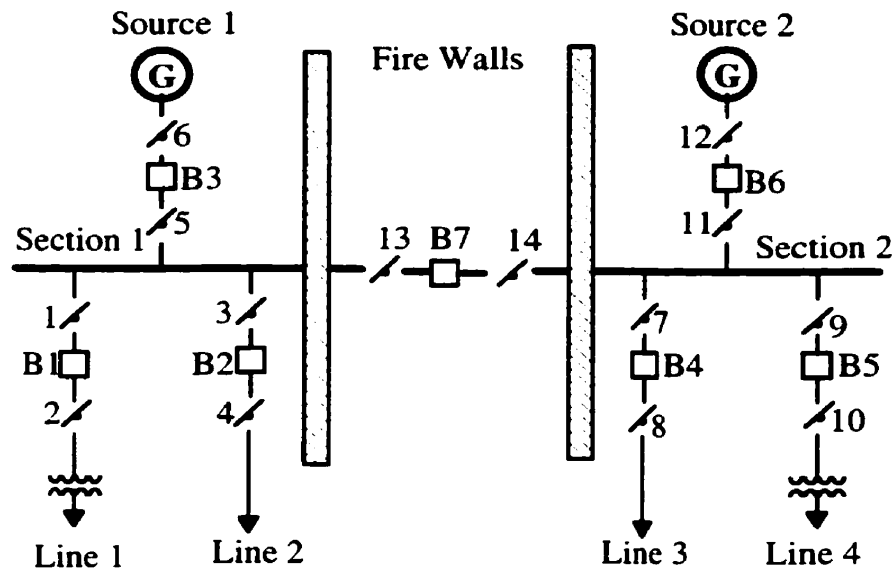
A single busbar substation is the least expensive substation arrangement. The disadvantages of using this configuration are that the entire substation is shut down when

- a busbar fails,

- a circuit-breaker fails, or
- maintenance is to be performed on the busbar.

### 2.3.2 Sectionalized single busbar arrangement

A sectionalized single busbar arrangement is similar to the single busbar arrangement except that the busbar is divided into sections which are connected by sectionalizing switches. A typical sectionalized single busbar arrangement is shown in Figure 2.5. The figure also shows fireproof walls that separate the sectionalizing circuit breaker from the bus sections to reduce the fire hazard. Before the two sections are connected to each other, the two systems must be synchronized. The sectionalizing circuit breaker, B7 shown in Figure 2.5, helps to synchronize the two sections when they are to be connected together.



**Figure 2.5.** A typical sectionalized single-busbar system.

If generator, Source 2 shown in Figure 2.5, is to be taken out of service, the following operations must be performed.

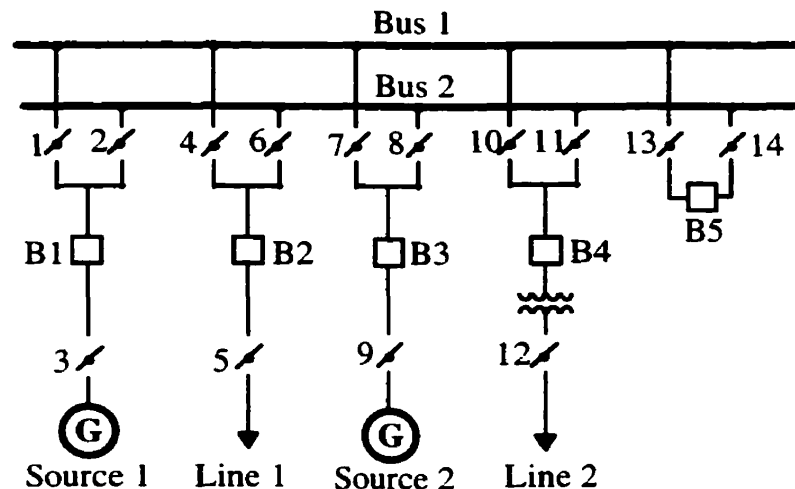
- Close isolators 13 and 14.
- Synchronize the two sections and close circuit breaker B7.

- Open circuit breaker B6.
- Open isolator 12.

This sequence of operations ensures that the supply to Line 3 and Line 4 is not interrupted when Source 2 is being taken out of service.

### 2.3.3. Double busbar arrangement

Major substations are usually equipped with double busbars to increase flexibility of operation and ease of maintenance. Normally, one set of busbars is in use while the other remains available for use in the event of a bus fault in the substation. The two busbars can be connected by a circuit that consists of a bus-coupler circuit breaker and two isolators. The bus-coupler circuit breaker is used to synchronize the systems connected to the two busbars. Each incoming and outgoing feeder can be connected to either busbar. A typical double busbar arrangement is shown in Figure 2.6.



BB = Bus-coupler circuit breaker

**Figure 2.6.** A typical double busbar arrangement.

This configuration makes it convenient to perform maintenance on busbars, test and commission new feeders, and operate a circuit at a different voltage. The disadvantage of

this arrangement is that all the circuits connected to a bus are interrupted in the event of a bus fault.

An appropriate sequence of operations must be used to transfer circuits from one busbar to the other. If a source, Source 2 shown in Figure 2.6, is to be brought into service while Bus 1 is energized by Source 1, the following operations must be performed.

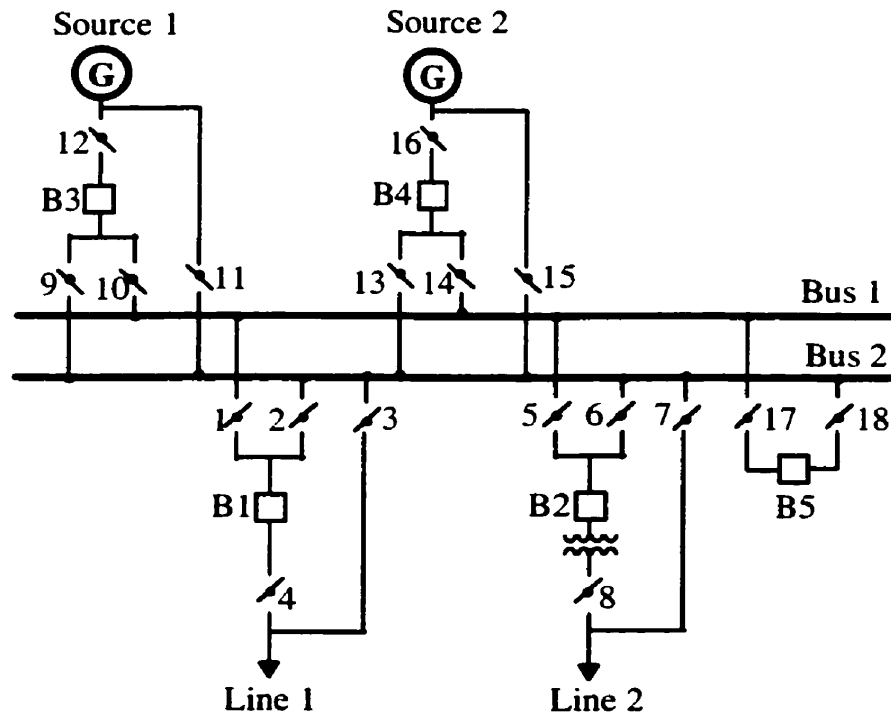
- Close isolators 8 and 9.
- Close circuit breaker B3, energizing Bus 2.
- Close isolators 13 and 14.
- Synchronize Bus 1 and Bus 2, and close circuit breaker B5.
- Close isolator 7, and then open isolator 8.
- Open circuit breaker B5.
- Open isolators 13 and 14.

#### **2.3.4 Double busbar with by-pass isolator arrangement**

The double busbar arrangement becomes extremely flexible when an additional isolator bypassing the circuit breaker is included. This arrangement is shown in Figure 2.7. The addition of a by-pass isolator improves the utilization of the circuit breakers. When a circuit breaker is to be taken out of service, the supply to the circuit can be maintained via the bus-coupler circuit breaker and the by-pass isolator.

If Line 1 is connected to Bus 1 via circuit breaker B1 and the circuit breaker is to be taken out of service, the following switching operations must be performed.

- Close isolators 17 and 18.
- Close circuit breaker B5 connecting Bus 2 with Bus 1.
- Close by-pass isolator 3.
- Open circuit breaker B1.
- Open isolators, 1 and 4.



**Figure 2.7.** A typical double busbar with by-pass isolator arrangement.

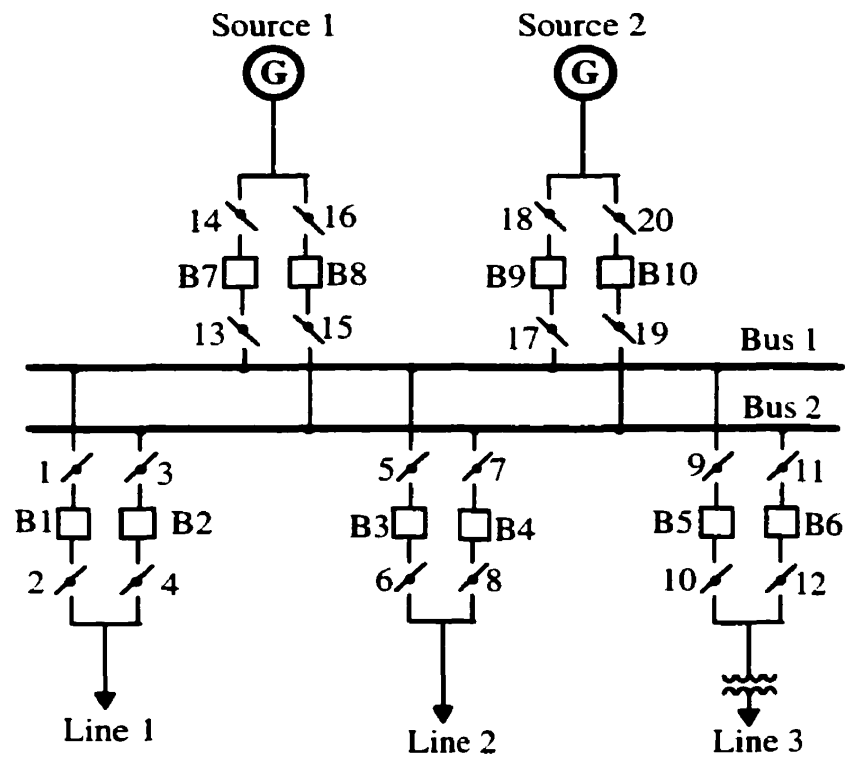
### 2.3.5. Two circuit breaker arrangement

In major substations, a scheme which has two circuit breakers in each circuit is sometimes used; this arrangement, called a double-bus-double-breaker scheme (or duplicate-busbar scheme), is shown in Figure 2.8. Because each circuit has two dedicated circuit breakers, a circuit can be connected to either busbar. Each circuit may be connected to both busbars.

This arrangement can provide un-interrupted power supply even if a fault is experienced on one of the busbars. It does not require a bus-coupler circuit breaker. Any busbar and any circuit breaker can be taken out of service for maintenance without interrupting power to the circuit. It is considered to be one of the best configurations for high voltage substations because of its flexibility, but it is also one of the most expensive alternatives.



If a fault is experienced on Bus 1, circuit breakers B1, B3, B5, B7, and B9 are opened to clear the fault.

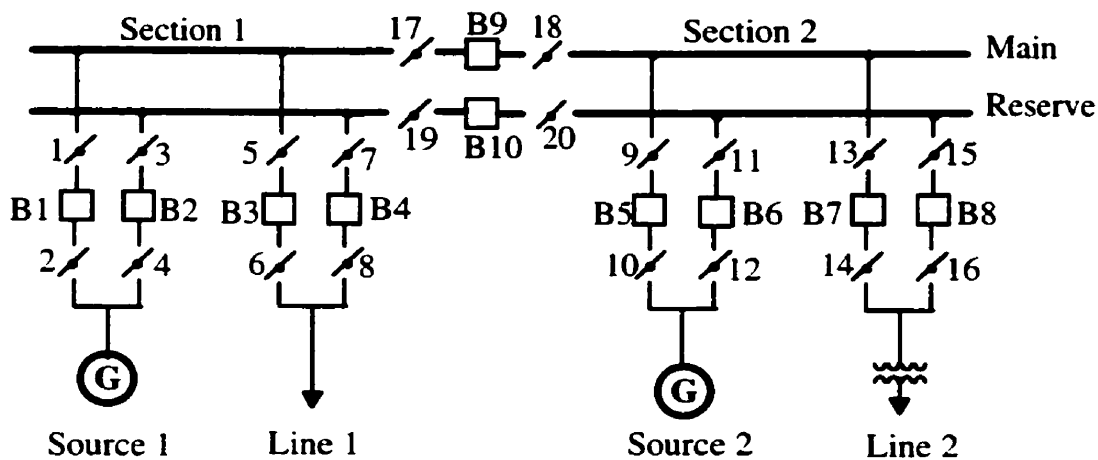


**Figure 2.8.** A typical two-breaker system.

### 2.3.6. Sectionalized two circuit breaker arrangement

This arrangement has two circuit breakers in each circuit as in the double-bus-double-breaker arrangement. The difference is that the busbars of the sectionalized duplicate busbar arrangement are sectionalized by using circuit breakers as shown in Figure 2.9.

This arrangement considerably enhances the flexibility of a duplicate busbar arrangement. Any section of the busbars can be isolated for maintenance. Sectionalizing of the reserve busbars is possible but is not always done. The cost increases due to the need for two sectionalizing circuit breakers. This makes the arrangement less attractive for most of the substations.



**Figure 2.9.** A sectionalized two-breaker system.

If Section 1 of the main busbars experiences a fault, circuit breakers B1, B3 and B9 are opened. If the busbar has to be taken out of service to repair the damage, isolators 1, 5 and 17 are opened.

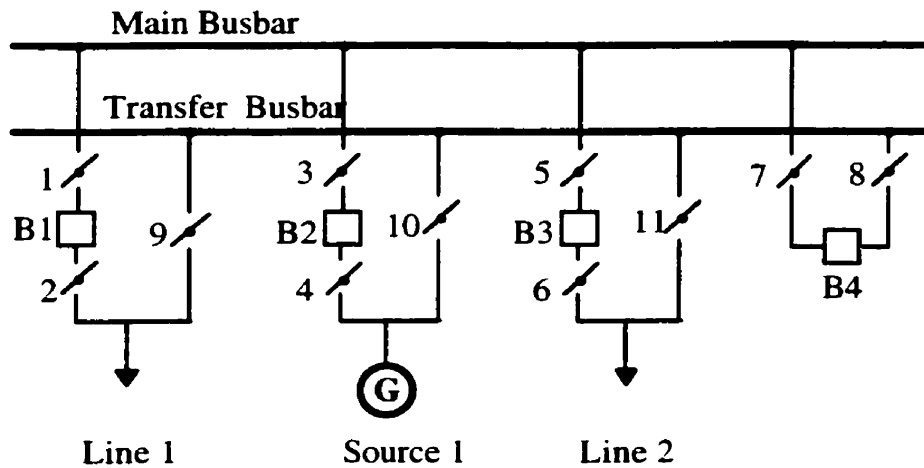
### 2.3.7. Main-and-transfer bus arrangement

A main-and-transfer bus arrangement incorporates by-pass isolators, as shown in Figure 2.10. When a circuit breaker between a line and the main busbar is out of service, the circuit controlled by it can be energized via the bus-coupler and the by-pass isolators. Protection for the circuit must be provided by the bus-coupler. This arrangement can afford to lose one circuit breaker at any one time without interrupting permanently the supply of power.

It is possible to take a circuit breaker out of service but not interrupt the supply of power on the circuit. For example, if circuit breaker B1 shown in Figure 2.10, is to be taken out of service, the following operations must be performed.

- Close isolators 7 and 8.
- Close busbar coupler B4 connecting the transfer busbar to the main busbar.
- Close by-pass isolator 9.
- Open circuit breaker B1.

- Open isolators 1 and 2.



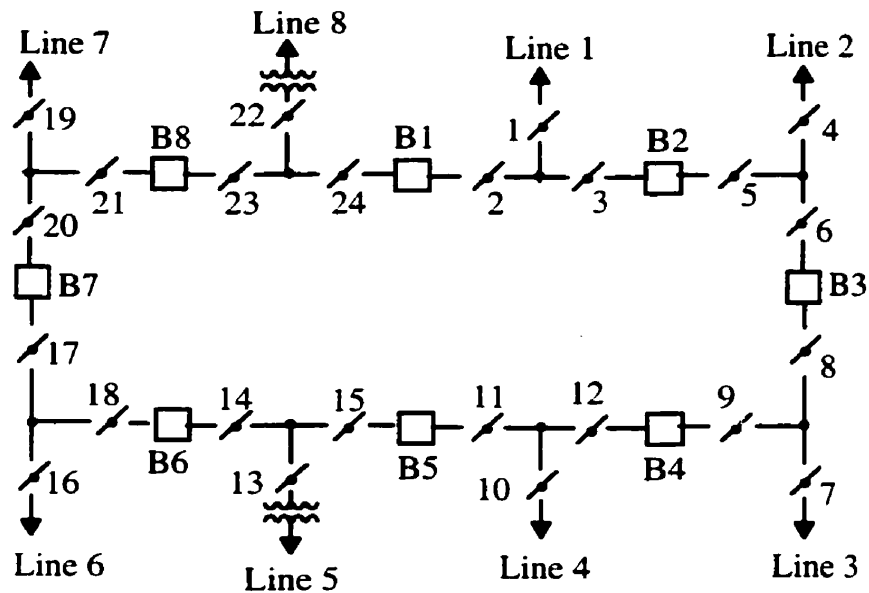
**Figure 2.10.** Main and transfer busbar arrangement.

### 2.3.8. Ring busbar arrangement

Some major substations use the ring (mesh) arrangement, called ring bus, shown in Figure 2.11. In this arrangement only one circuit breaker is required per circuit but a circuit breaker can be taken out of service without interrupting the supply of power to any line. Two circuit breakers must be opened to take a circuit out of service. This is a very widely used arrangement because of its flexibility and low cost.

This arrangement has a low initial and ultimate cost and provides flexibility in operation for maintenance of breakers. The arrangement, however, makes protective relaying and automatic reclosing somewhat complex.

When a fault is experienced in Line 1, shown in Figure 2.11, circuit breakers B1 and B2 are opened to clear the fault.



**Figure 2.11.** A ring busbar arrangement.

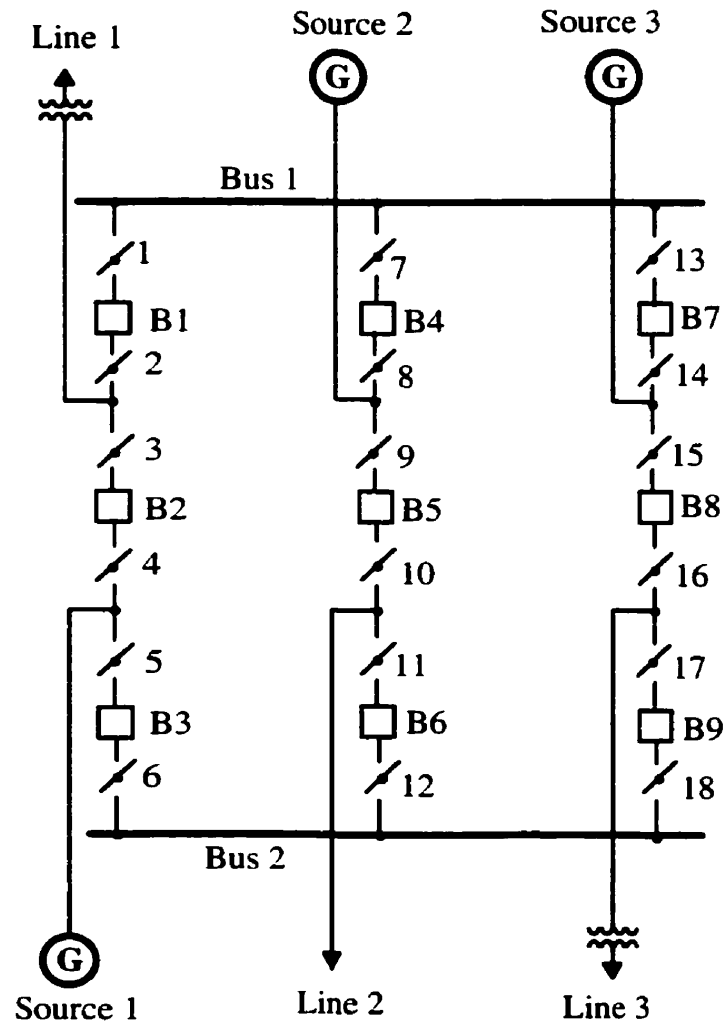
### 2.3.9. Breaker-and-a-half arrangement

A breaker-and-a-half arrangement is a hybrid of a ring busbar arrangement and a double-bus double-breaker arrangement. The switching arrangement is shown in Figure 2.12. As the name implies, it requires one and a half circuit breakers per circuit because each pair of circuits is controlled by three circuit breakers. This arrangement has been one of the most popular arrangements in recent years.

This looks partly like a ring busbar arrangement, and it is in essence a combination of several ring like arrangements rather than a single ring. This arrangement makes it easier to use switches with lower continuous current ratings than that in a ring busbar arrangement. Two lines sharing the circuit breakers are usually arranged such that one is connected to a source and the other to a load. This source-load combination minimizes the flow of current on the busbars and the switches.

This arrangement is less expensive than a two circuit breaker arrangement but provides comparable overall performance. The flexibility and reliability are high, but relaying and

reclosing are not easy to implement.

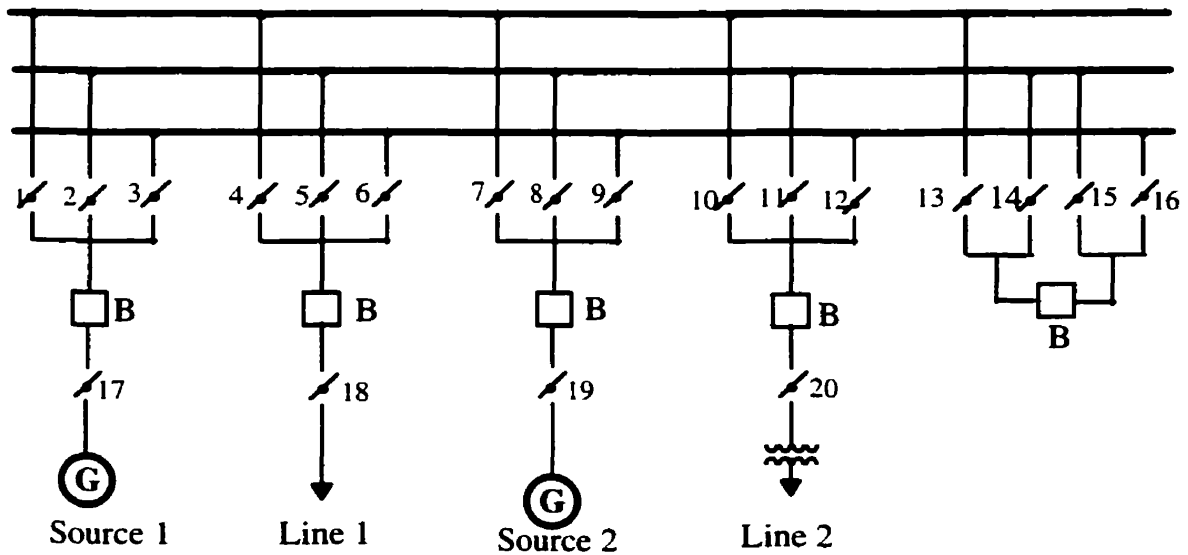


**Figure 2.12.** A breaker-and-a-half system.

### 2.3.10. Multiple bus arrangement

This arrangement, shown in Figure 2.13, consists of more than two sets of busbars. Any bus can be isolated for maintenance. A circuit can be readily transferred from one busbar to another using the busbar-tie breaker and bus selecting switches. However, it is not a popular arrangement because it is not cost effective and its performance does not surpass the two-circuit-breaker, breaker-and-a-half and ring arrangements. In this arrangement, a

circuit breaker cannot be serviced without interrupting power supply to a line.



**Figure 2.13.** A multiple busbar system.

## 2.4. Summary

Power systems are monitored and controlled from substations using switching and control equipment. The switches and other equipment used in substations have been introduced in this chapter and their typical roles have been described. The commonly used busbar configurations have been described and their advantages and disadvantages have been outlined.

Some of the other arrangements, which are not discussed in this chapter, include the three circuit breaker arrangement, the breaker-and-a-third arrangement, and the single and double or multiple busbars with by-pass bus arrangement [3-6].

### **3. SUBSTATION SWITCHING**

Switching operations are performed to rearrange the topology of a power system so that various operating requirements of the system are met. The usual switching operations are outlined in this chapter. The roles of the operations and the importance of computer based interlocks are discussed. Important publications concerning the topics are also reviewed.

#### **3.1. Types of Switching Operations**

Switching operations are performed in response to randomly occurring and scheduled events. Some examples of the events that require switching operations to be performed are as follows.

- i. **Faults:** Isolate the faulted equipment and restore the power supply to healthy circuits.
- ii. **Overloading:** Redistribute loads so that they are fairly distributed among the circuits.
- iii. **Overvoltage or undervoltage:** Reactors are switched on or off, or transformer taps are changed to bring the bus voltage within limits.
- iv. **Maintenance:** Equipment is isolated and then grounded before maintenance is performed.
- v. **Restoration:** When part of a circuit goes out of service, the network is reconfigured such that the customers are not deprived of power supply.

Some of these events, such as faults, can cause serious damage and must, therefore, be cleared without undue delay. In addition to equipment damage, system instability can occur if faults, or disturbances, are not cleared promptly. Dedicated relaying systems are, therefore, used to open appropriate circuit breakers to disconnect the faulted section from the system. In some cases, relaying systems are designed to re-close the circuit breakers to restore power supply in case the faults were of a temporary nature. If a fault is of a

permanent nature, the flow of power is rerouted around the faulted equipment.

Some events, such as minor overload, and over voltage / undervoltage conditions, occur gradually and short delays in decision making can be permitted. A detailed analysis of the state of the system and the consequences of subsequent switching operations can, therefore, be conducted before implementing these switching decisions.

The ability of a power system to withstand the outage of an equipment can be determined by performing security analysis. If the analysis shows that the outage is likely to result in overloading of some components, the system can be reconfigured to prevent overloads.

When it comes to scheduled maintenance, a system can be analysed before any switching operation is performed. The equipment is isolated before maintenance work is performed.

Considering the foregoing discussion, switching operations in a substation can be divided into the following four categories:

- i. Fault switching,
- ii. Restoration switching,
- iii. System re-configuration switching, and
- iv. Maintenance switching.

Among these four categories, the fault switching is the most time critical and the maintenance switching is the least time critical. This thesis concentrates mostly on the system re-configuration and maintenance switching.

The discussion of Section 2.2 shows that all equipment and switches are not robust devices like circuit breakers. Therefore, the operation of circuit breakers and other switches must be coordinated. Utilities employ special safety features to prevent undesired switching operations from being performed inadvertently.

### **3.2. Interlocking**

Restrictions are placed on the sequence in which switches can be operated in a substation to ensure security of the system and safety of personnel. These restrictions are usually



referred to as interlocks.

Interlocking operations can be classified into two groups, operational interlocking and maintenance interlocking. Operational interlocking prevents switching operations that can cause damage to equipment or can cause inadvertent loss of power to the customers. Maintenance interlocking protects personnel working on the equipment. These interlocks are more restrictive and include the use of padlocks in addition to usual interlocks.

It used to be a utility practice to mechanically interlock the operation of isolators, earth switches and enclosures of equipment. For example, a circuit breaker and its adjoining isolators were mechanically interlocked so that the isolator could not be operated unless the circuit breaker was opened. The mechanical arrangements included camshafts, bolts and padlocks. These schemes took substantial time to lock and unlock the switches.

Mechanical interlocks are still in use. They are now used mainly to allow safe access for personnel to carry out maintenance, or to prevent access to the enclosures containing live equipment.

Newer substations use electrical schemes, where the present status of the switches and the desired switching actions are supervised by logic controllers. The safety of the switching operations is ensured by the preprogrammed logic.

Electrical interlocks are superior to mechanical interlocks because:

- The components do not have to be physically adjacent in order to apply electrical interlocks.
- They permit an interlock of any given switch to the state of any number of components of the substation.
- They can supervise the state of the system in addition to checking the state of the switches.
- They make it possible to perform sequence switching irrespective of the complexity of a substation.

These superior features, however, increase the complexities of the design of the switching logic.

A combination of electrical and mechanical interlocks can provide a comprehensive interlocking system.

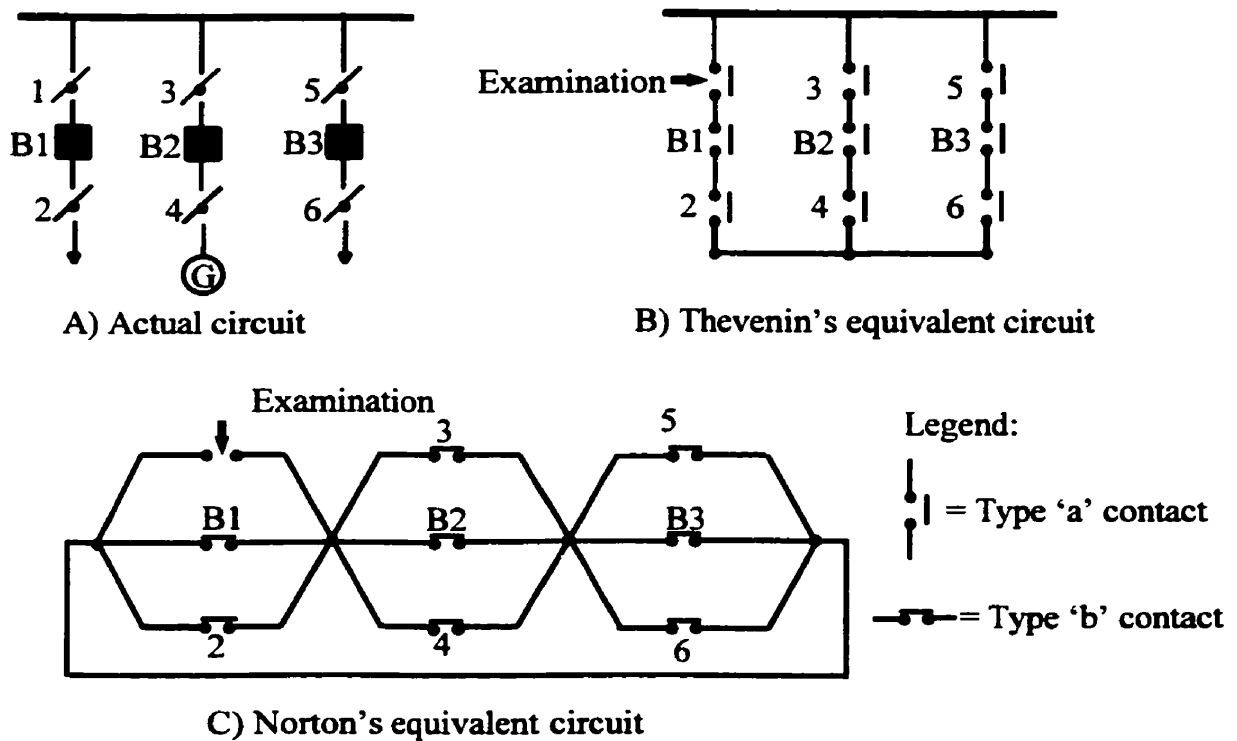
### **3.3. Developments in Interlocking Techniques**

In the early days of power systems, only simple deterrents, such as padlocks, were provided to prevent unauthorized operations that might be incorrect. The utilities relied on the vigilance of operators in performing switching sequences correctly. Interlocks were introduced to reduce the chances of inadvertent switching errors. Later, some industries implemented overly restrictive interlocking arrangements to cover all possible contingencies. More compromising solutions were then sought to provide comprehensive interlocks without making them overly restrictive.

Two notable publications in this area are those of Hutchinson [7] and Cory [8]. Hutchinson suggested the use of two analog circuits, which represent the interlock logic for the isolators of a substation. One circuit blocks the operation of an isolator if it makes or breaks load currents and the second circuit allows the operation of an isolator if a low impedance shunt path exists across it. The first circuit represents the Thevenin's equivalent of the system in which all impedances are neglected. The second circuit represents the Norton's equivalent of the system in which all admittances are neglected.

The first circuit is made by connecting the type 'a' auxiliary switches of the circuit breakers and isolators in the configuration of the substation. For the substation of Figure 3.1(a), the circuit is shown in Figure 3.1(b). If appropriateness of operating an isolator is to be checked, the impedance across its auxiliary contact is measured. The operation is allowed if the measured impedance is infinity. In case the impedance is zero, the second circuit is used to make the final decision.

The second circuit is made by connecting the type 'b' auxiliary switches of the circuit breakers and isolators in the Norton's equivalent configuration of the substation. For the substation of Figure 3.1(a), the circuit is shown in Figure 3.1(c). If appropriateness of operating an isolator is to be checked, the impedance across its auxiliary contact is measured. The operation is allowed if the measured impedance is infinity.



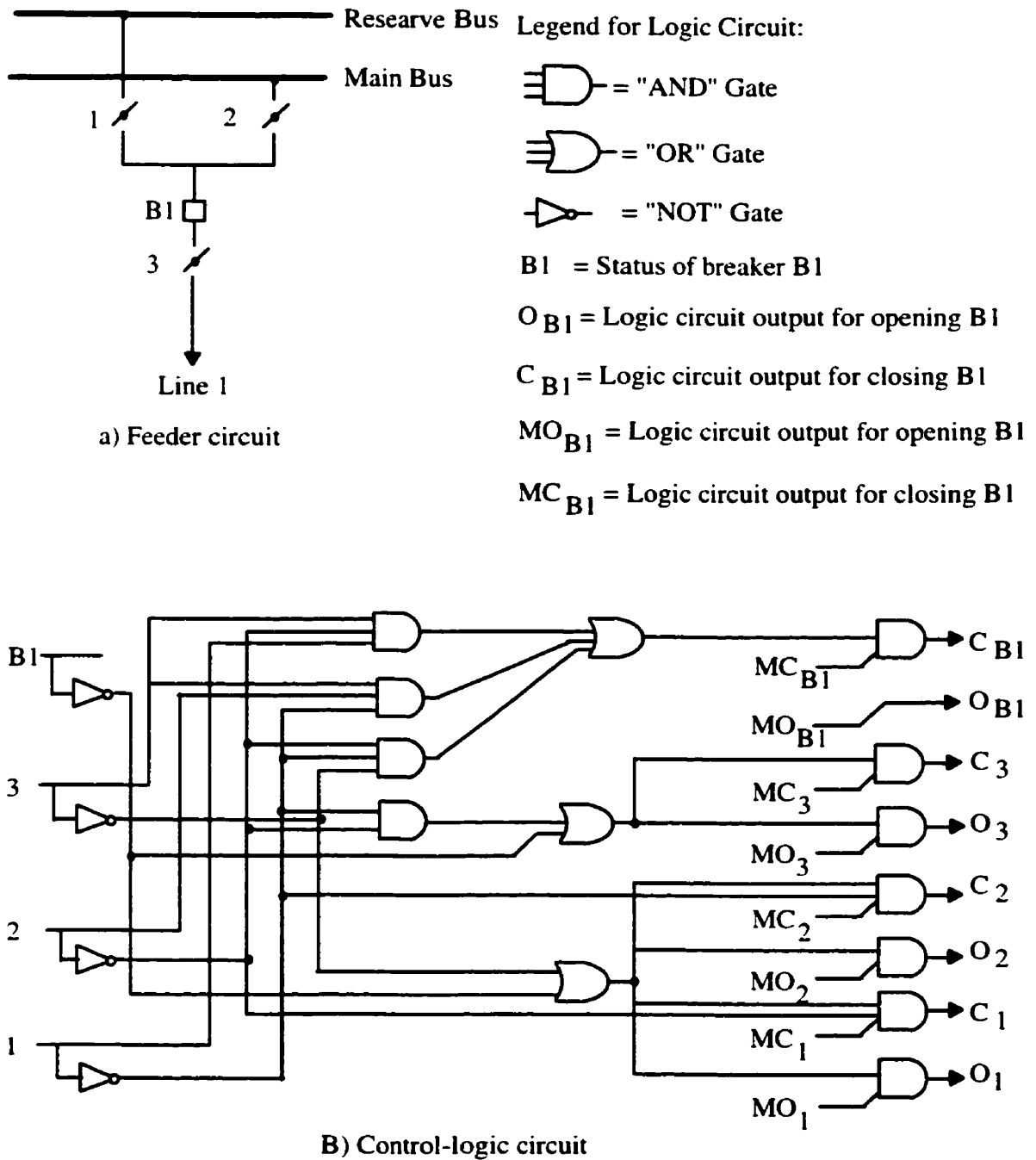
**Figure 3.1.** A single busbar substation and its equivalent circuits.

Cory [8] represented switching constraints by using Boolean algebra and implemented the logic in a hardware circuit. This work shows that complex logic can be implemented with electronic controllers.

A logic circuit prepared by Cory for a double-busbar substation is shown in Figure 3.2. The inputs to the control circuit are the status of the switches and the commands to operate them. The closing or opening of a switch is permitted only when the output from the controller is logical TRUE.

Hope and Cory [9], Row and Cory [10] extended the work reported in Reference 7. More

notable of these works is Reference 9. Hope and Cory simplified the interlock logic off-line and generated appropriate Boolean logic by using digital computer programs.



**Figure 3.2.** A feeder circuit and an interlock logic circuit for the feeder circuit.

Couch and Morrison [11] proposed a different approach in designing interlocking schemes. Rather than finding all switching combinations off-line and storing them, they proposed the use of a digital computer program to determine on-line the effect of a switching operation on the safety of the system. They analyze the substation topology, status of the switches and live/dead status of the lines to determine the impact of a switching operation on the security of the substation. Figure 3.3 shows a flowchart, taken from Reference 11, that is used to check the impact of operating a switch.

Lidgate et al. [12] used a network-tracing (graphical) technique to design the generalized interlocks. The logic was described using logic diagrams and was translated into two tables, a “decision table” and an “anti-table.” The decision table contained a list of all the combination of switches that prevent the operation of a switch. The anti-table listed every combination of switches that allow the operation of a switch.

Brand, Kopainsky and Wimmer [13] proposed a topology-based interlocking technique which decouples interlocking rules and substation configurations. This technique provides interlocking schemes that can implement the rules irrespective of the configuration of a substation.

### **3.4. Sequence Switching**

Sequence switching consists of selecting the most appropriate intermediate steps when the network configuration is to be changed. Almost all switching operations described in Section 3.1 require that the switches be operated in a proper sequence.

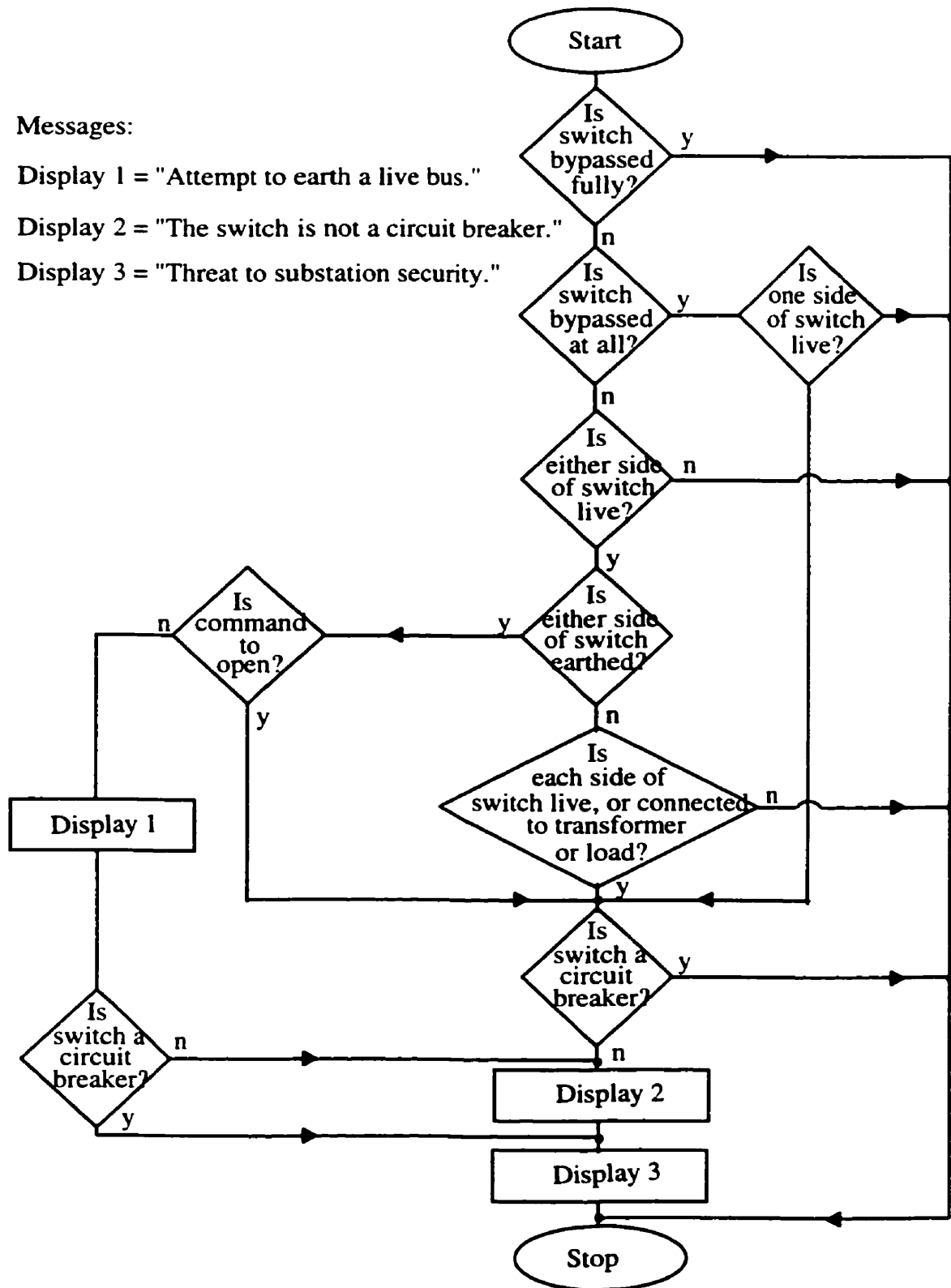
A properly planned switching sequence must change the network configuration with a minimum number of switching operations. The sequence must not violate the switching constraints, cause loss of power supply or endanger the safety of personnel and the equipment. A switching sequence may consist of a combination of opening and closing of switches. Determining a switching sequence can be a complex process because several constraints may have to be satisfied.

Messages:

Display 1 = "Attempt to earth a live bus."

Display 2 = "The switch is not a circuit breaker."

Display 3 = "Threat to substation security."



**Figure 3.3.** A flowchart for checking the security of a substation.

Except for the protection and restoration switching done by dedicated relays, switching plans are prepared by the substation personnel. It, however, takes time to manually generate appropriate switching sequences. A computer based system with appropriate features would allow engineers to program the interlock requirements of a substation, and generate and evaluate the switching sequences. Such a system would save time and help reduce human errors in decision making. If the technique and its applications are reasonably inexpensive and user friendly, they would be useful to substation engineers, and would be readily accepted and applied.

### **3.5. Developments in Sequence Switching**

Hope and Cory [8] used a technique in which they stored all permissible switching combinations in a “primitive flow table” in an ordered form. The ordering in the table is such that the adjoining rows differ by the state of one switch that can be operated safely. To change the configuration of a substation, switches are opened and closed as dictated by the transition of states between adjoining rows. The procedure starts from the row that matches with the initial state and stops at the row that matches with the final state. Other tables are also created to make it easier to analyse the information in the primitive flow table.

Couch and Morrison [11, 14] proposed a method of selecting optimal switching sequence based on dynamic programming. This method minimizes an objective function which consists of the cost of switching operations. The technique was used for off-line selection of switching sequences. The main difficulty with this approach is that it uses a penalty cost associated with each switching operation which is difficult to determine.

Udo [15] also proposed an optimization technique for selecting switching sequences. He showed that most criteria result in selecting a sequence that consists of the least number of switching operations. He used this approach instead of the penalty functions and cost minimization.

Traca-de-Almeida [16,17] proposed a method different from the dynamic programming

based techniques, which scan a large number of network configurations where the combinations to be considered grow exponentially with the size of the system. In the proposed method, the initial and final configurations of the substation were used for determining the switching sequence. The procedure consisted of the following steps:

- Close isolators that must remain closed in the final configuration.
- Close circuit breakers that must remain closed in the final configuration.
- Open isolators that must remain open in the final configuration.
- Open circuit breakers that must remain open in the final configuration.

For example, in a substation whose initial configuration is as shown in Figure 3.4(a) and the final configuration is as shown in Figure 3.4 (b), application of the suggested procedure provides the following switching sequence.

- Close isolator 4.
- Close isolator 8
- Close isolator 11.
- Close isolator 9.
- Make synchro-check and close circuit breaker B12.
- Close circuit breaker B10.
- Make synchro-check and close circuit breaker B4.
- Make synchro-check and close circuit breaker B8.
- Open circuit breaker B3.
- Open isolator 3.

Weatherall and Senior [18] designed a microprocessor control system for post fault sequence switching. In their scheme, status of relays and switches are entered in the form of a bit pattern which is compared with the patterns stored as tables in the computer memory. The sequence of operations associated with the matched pattern is implemented. If no matching bit pattern is found, an alarm is generated and automatic switching is blocked. Other notable works done in post-fault sequence switching are those of Marlow and Dauncey [19], and Dialynas and Machias [20].

The expert system based techniques provide some scalability in the development of



switching schemes. They normally start with a simple and workable solution and provide gradual refinement to the system as more experience is gained and the knowledge about the system expands. The next section describes some expert system based developments that influenced the work reported in this thesis.

### **3.5.1. Use of expert systems**

Some [21-23] proposed the use of an expert system for sequence switching. In the work of Zhang, Hope and Malik [22], they broke the complex tasks into simple sub-tasks. They used the criterion of minimizing the number of switching operations when selecting optimal switching sequences. They also included a provision to incorporate the cost minimization criteria if the user can provide information on the cost of switching operations.

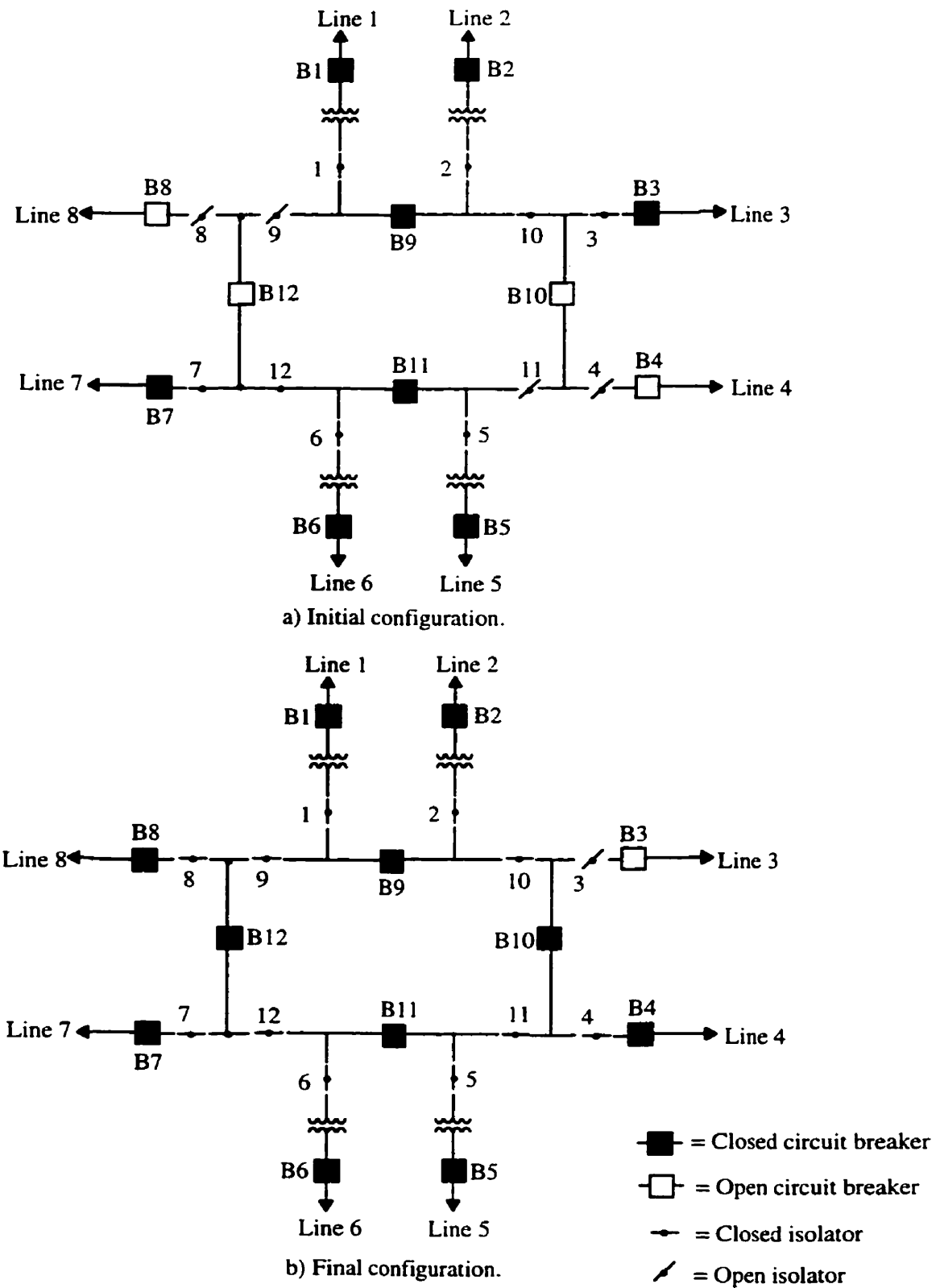
In this work, the elements of network graphs are grouped into three categories: components, nodes and branches. The switching operations are grouped into five categories. Alternate sequences of switching operations are determined by searching the network graph for alternate paths. “Best-first-search” technique is used to expedite the search.

Schulz and Wollenberg [23] also developed an expert system to generate and evaluate switching sequences for equipment maintenance. They included power flow calculations in selecting optimal switching sequences. They divided the expert system into two sub-systems, “switching sequence creation expert system” and “switching sequence evaluation expert system.” The first system determined

- the switches that form isolation boundary for the equipment to be taken out of service,
- switching steps for isolation, and
- switching steps for restoration.

The second system reads a switching sequence and evaluates its correctness and optimality.

Other publications [24-27] that deal with the use of expert system in substation switching have not been discussed in this chapter because they do not seem to make significant contributions to the work reported in this thesis.



**Figure 3.4.** Initial and final configurations of a substation.

### **3.6. Summary**

Switching actions in a substation are carried out to protect the equipment from system events such as faults and disturbances. Operations are also carried out to prepare the system for repair and maintenance work.

Switches are interlocked so that they can operate only when they do not pose any threat to the security of the system or to the safety of personnel. Switches are, therefore, operated sequentially so that the interlocking rules are obeyed.

The configuration of a substation can be changed by implementing alternative switching sequences. The sequence that consists of minimum switching operations is generally preferred.

## **4. SWITCHING DATA AND PROCEDURES**

The components used in a substation have many common characteristics. The characteristics can be analysed and then used to develop operating constraints taking the design and placement of the components in the substation into consideration. The steps involved in determining switching sequences, and the factors to be considered in that process are discussed in this chapter. The operating constraints of the equipment used in substations are described. An approach, which groups interlocks in four categories, to simplify and expedite the implementation of interlocks is presented.

### **4.1. Basic Switching Procedures**

A basic switching procedure may consist of an operation of a switch, or a series of closing and opening operations. This section describes basic switching procedures. Some complex switching procedures, composed of a number of basic operations, are described later in Section 4.2.

#### **4.1.1. Opening and closing of a switch**

Some switching procedures consist of opening and closing of a switch. Typical examples include grounding of an isolated component or opening a circuit. Opening (or closing) a switch also forms the fundamental component of switching procedures such as connecting a line to a busbar. A closing or opening operation of a switch is carried out only after ascertaining that the component level, substation level and system level security interlocks are not violated.

#### **4.1.2. Bypassing a component**

Certain situations require that a component be bypassed. For example, a circuit breaker may have to be bypassed for performing maintenance on it. To maintain the continuity of power supply, a path around the circuit breaker may have to be established. The component, substation and system level interlocks must be satisfied when these operations are to be performed.

#### **4.1.3. Operating switches in a branch**

Branches can contain multiple switches such as a circuit breaker and isolators. When a branch contains an isolator and a circuit breaker, they must be operated in a definite sequence. When opening a branch, the circuit breaker must be opened before the isolator is opened. When closing a branch, the isolator must be closed before the circuit breaker is closed. The operation of a branch that contains only isolators must be coordinated with the operation of its adjoining branches that contain circuit breakers. The component, substation and system level interlocks must be satisfied when a branch is opened or closed.

#### **4.1.4. Connecting a line to a busbar**

Often, more than one branch is involved in connecting a line to a busbar. It is desirable that the closing of switches progress from the source side to the load side when a line is to be connected to a busbar. Anyhow, isolators must be closed before the circuit breakers are closed. All levels of interlocks must be checked before any operations are performed. When two live circuits are to be connected, they must be synchronized.

#### **4.1.5. Disconnecting a line from a busbar**

Disconnecting a line from a busbar often involves the opening of a circuit breaker between the line and the busbar. In some circumstances, it may require the opening of all switches in the power-flow path(s) between the line and the busbar. The circuit breakers must be opened before the isolators are opened. It is desirable that the opening of switches progress from the load side to the source side when a line is to be disconnected from a bus. All levels of interlocks must be checked before disconnecting a line.

#### **4.1.6. Connecting busbars**

Normally, the sections of a busbar are connected together via bus sectionalizing switches. Occasionally, however, the sections are operated in stand-alone modes. When two disconnected but live busbars are to be re-connected, the busbars must be synchronized before closing the sectionalizing switches. If synchronizing facilities are not available, one of the buses must be de-energized before they are connected.

In the case of a double bus scheme, the lines are not connected to the main and auxiliary buses through separate circuit breakers, but rather they have one circuit breaker per line. When transferring circuits from the main bus to the auxiliary bus in a double busbar scheme, the buses must be connected via a bus-coupler circuit breaker.

### **4.2. Advance Switching Procedures**

Bus transfer, partitioning of a substation, isolation of a circuit, load shedding and restoration, and maintenance of a busbar are examples of complex switching procedures. These procedures consist of several basic operations that can be divided into sub-tasks.

#### **4.2.1. Identifying equipment to be taken out of service**

To identify the equipment that needs maintenance, maintenance schedules should be checked. To identify the equipment for repair, monitoring system and visual-inspection reports should be checked.

#### **4.2.2. Determining isolation boundary**

A component must be properly isolated and discharged before maintenance work is started. Every equipment adjacent to the component to be repaired must be taken out of service. Once a component is isolated, all adjacent components must be disabled so that they are not operated inadvertently until the repair is done and the personnel have released the equipment.

#### **4.2.3. Identifying affected circuits**

Some loads served by a substation may be curtailed if a component is isolated without providing alternate route to deliver power. Moreover, this may overload an equipment that is connected in parallel with the component being taken out of service.

#### **4.2.4. Rerouting flow of power**

If overloading of an equipment or a curtailment of load is imminent, alternate routes to feed the loads should be determined. If such an alternative does not exist, the emphasis should be on providing power supply to as many loads as possible.

The following steps should be performed while determining the switching sequence for restoring power supply.

1. Exclude the switches and equipment that are blocked.
2. If necessary, connect circuits through alternate routes.

#### **4.2.5. Isolating an equipment**

The switching sequence for isolating an equipment is prepared from the topology of the substation. The following operations should be performed for this purpose.

1. Check if all the circuit breakers in the sequence can be opened without violating the security of the system.
2. Check if all the isolating switches in the sequence can be opened without violating the security of the system.
3. Open all the circuit breakers in the list.
4. Open all the isolators in the list.

While performing these steps, process should abort if the test produces unfavourable results or if a switch fails to operate.

#### **4.2.6. Grounding an equipment**

An equipment is grounded after it is confirmed that it has been isolated. If a grounding

switch exists within the isolated section of the circuit, the equipment should be discharged by closing the grounding switch. The unavailability of a grounding switch within the isolated part of the circuit would require that the component must be grounded by using a mobile grounding unit.

#### **4.2.7. Restoring circuits**

The operations that were performed to isolate the circuit are known. Therefore, it is relatively straight-forward to restore the circuit to its original configuration. The following steps should be performed for restoring a circuit.

1. Open the ground connections.
2. Check if all the isolating switches, which must be closed in the final configuration, can be closed safely.
3. Check if all the circuit breakers, which must be closed in the final configuration, can be closed safely.
4. Close the isolators identified in Step 2.
5. Close the circuit breakers identified in Step 3.
6. Check if all the circuit breakers, which must be opened in the final configuration, can be opened safely.
7. Check if all the isolators, which must be opened in the final configuration, can be opened safely.
8. Open the circuit breakers identified in Step 6.
9. Open the isolators identified in Step 7.

### **4.3. Formulation of Constraints**

The operating state of the system and characteristics of the switches must be analysed to determine the validity of an intended switching operation. The design of the equipment, operating parameters, and constraints necessary to operate the system safely must be considered [28, 29]. The constraints can be expressed in a general form and applied effectively to substations of all practical configurations.



#### **4.3.1. General**

An analysis of operating practices of substations dictates that the following requirements must be met by a switching operation.

- Protection systems must be allowed to open circuit breakers to clear faults.
- The most recent status of the network must be available.
- The current and voltage ratings of the lines, transformers and other equipment must not be exceeded.

A practical switching device is different from an ideal on-off device in two ways. Firstly, its closing and opening times are not negligible. Secondly, its status may not be known. A switch must go through the following interlock constraints:

- Only one switching operation should be permitted at a time.
- Some switches must not be operated before the operation of another switch is completed.
- A switch must not be operated if its status is unknown.

#### **4.3.2. Circuit breakers**

Given the voltage, power factor and impedance of different parts of the system and the system topology, the maximum possible fault current that a circuit breaker must interrupt can be calculated. The breaking capacity of a circuit breaker should always be greater than or equal to the maximum possible fault current. A circuit breaker must satisfy several constraints for it to be safely opened or closed.

##### ***4.3.2.1 Opening Operation***

- A circuit breaker must not be opened if it inadvertently interrupts power to loads or causes overloading of parallel circuits.

The possibility of loss of power to customers due to the opening of a line breaker can be checked by analysing the topology of the network. The possibility of overloading of lines following the opening of a circuit breaker can be determined from a power flow analysis of the system.

#### **4.3.2.2 Closing Operation**

- A circuit breaker must not be closed if its adjoining isolator is in motion.
- Two live lines must not be connected without synchronizing them.
- A live line must not be grounded.

The closing of a circuit breaker while its adjoining isolator is in motion would set up an arc across the isolator contacts and damage them severely. This requirement determines the minimum time required for completing a switching sequence.

If two live lines are connected while the difference between the magnitudes and phase angles of their voltage waveforms are beyond specified limit, large amount of power flows from one line to another, which could cause the system to loose stability. It is, therefore, necessary to synchronize the lines before connecting them.

Grounding of a live circuit initiates a ground fault resulting in the flow of excessive currents. This could damage the equipment and jeopardize the safety of personnel involved in grounding. A circuit breaker must not be closed if one of its terminals is connected to a source and the other is connected to ground.

#### **4.3.3. Load-break switches**

The constraints applicable to circuit breakers are also applicable to load-break switches. The load-break switches have one extra constraint as follows:

- A load-break switch must not break fault current.

This constraint is due to the design limitations of a load-break switch discussed in Section 2.2.2.

#### **4.3.4. Isolators**

- An isolator must not make or break load or fault current.
- On-load operation of an isolator must be permitted if there is a low impedance shunt path across it.

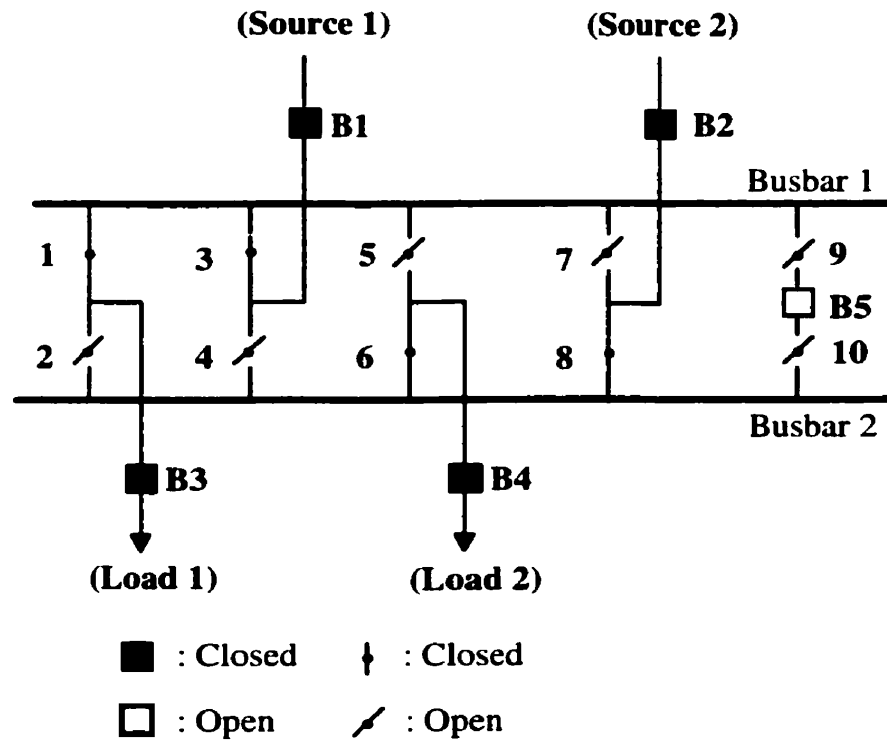
The constraints are due to the design limitations of an isolator that are discussed in Section 2.2.3. If isolator 5 of Figure 4.1 were to be closed, an arc would be struck across the

contacts of the isolator because of the difference between the potentials of two buses. This would severely damage the isolator. To avoid the damage, a shunt path across isolator 5 can be established by closing 9, 10 and B5.

#### 4.3.5. Grounding switches

- A live line must not be grounded.

An equipment must be properly isolated before grounding it. Once an equipment is grounded, the grounding circuit must be opened before energizing the equipment.



**Figure 4.1.** A substation with bus coupling circuit.

#### 4.3.6. Transformers

- Two transformers operating in parallel must be at identical tap settings.

Operating two transformers in parallel while their tap settings are different would cause circulating currents to flow between the transformers. These currents could damage the

transformers seriously.

#### **4.3.7. Sequence switching**

Switches are operated sequentially in substations. Only one switch is operated at a time to ensure that the general constraints are not violated. When this procedure is followed, it becomes easy to identify the location of a fault. Other constraints that must be met while implementing a switching sequence are as follows.

- Power supply must not be unnecessarily interrupted during and after the switching operations.
- Busbars or lines must not be de-energized and then re-energized during a switching sequence.
- Closing of a circuit breaker should not create a configuration in which the fault levels exceed the ratings of the circuit breakers.
- Redundant switching operations should be avoided.
- Among all the alternative switching sequences, the most economical one should be selected. It tends to be the shortest sequence and therefore may also assure restoration of supply in the shortest time.
- While restoring power to a line, the switches nearest to the source should be closed first and the switches farthest from the source closed last.
- The switching sequence should not violate the interlocking scheme.

#### **4.3.8. Substation specific constraints**

Despite having many common characteristics, every substation has a unique switching arrangement, importance and placement in the system. Some additional switching constraints might be necessary to meet the specific requirements of a system. The following are some examples of special constraints.

- An auxiliary bus must not be operated in parallel with the main bus except when circuits are being transferred between the main bus and the auxiliary bus.
- No more than two transformers should be connected in parallel in a sequence of operation [18].

## **4.4. Data Requirements and Data Structures**

System components must be represented by numeric data for use in a software application for substation control. The data should be kept in a database organized in a predefined structure. This would ensure efficient sharing and manipulating of data by different programs.

Many forms of databases are used by electric power utilities and manufacturers for keeping information on the equipment used in power systems. The data applicable to the development of substation switching schemes identified and adopted in this project are summarized in this section.

### **4.4.1. Switches**

The structure shown in Table 4.1 was prepared and used for storing data of the switches for this project. All the information in this data structure, except the 'Branch', is provided by the user. The 'Branch' is assigned by the software application, and is made available to the user upon request. All data in this structure are invariant except for the 'Status' of the switch, which is updated continually during on-line operation.

The status field can contain slightly different information for different types of switches. For example, a circuit-breaker may need status fields to indicate if its two ends are live and synchronized. The 'Relations' field lists the names of equipment with which the operation of the device is directly linked. For example,

- an isolator may be associated with a circuit breaker.
- a circuit breaker may be associated with a feeder, transformer or a busbar.
- a grounding switch may be associated with an isolator.

<b>Table 4.1. Switching device data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>ID</b>	<b>I</b>	<b>Identification number</b>
<b>Name</b>	<b>S</b>	<b>Name (any length)</b>
<b>Type</b>	<b>I</b>	<b>Type of switch: Circuit breaker, load-break switch, isolator, grounding switch, fuse switch</b>
<b>Current</b>	<b>F</b>	<b>Current rating, kA</b>
<b>Voltage</b>	<b>F</b>	<b>Voltage rating, kV</b>
<b>Link1</b>	<b>I</b>	<b>ID number of the link on one side of the switch</b>
<b>Link2</b>	<b>I</b>	<b>ID number of the link on the other side of the switch</b>
<b>Status</b>	<b>CD</b>	<b>Status of the equipment; open, closed, stuck, free, undefined, known, healthy, unhealthy, remote-controllable, manual</b>
<b>Branch</b>	<b>P</b>	<b>Pointer to the branch in which it belongs</b>
<b>ClosingT</b>	<b>F</b>	<b>Closing-time</b>
<b>OpeningT</b>	<b>F</b>	<b>Opening-time</b>
<b>Relations</b>	<b>SL</b>	<b>List of names of associated device separated by commas</b>
<b>Lock</b>	<b>CD</b>	<b>A complex structure, which defines when it is free to operate</b>
		<b>Legend:</b> <b>I = Integer</b> <b>P = Integer pointer</b> <b>F = Floating-point,</b> <b>S = Text string (Alphanumeric name)</b> <b>SL = List of text strings,</b> <b>CD = Complex data structure having many fields within</b>

#### 4.4.2. Nodes

Data of the nodes have to be kept in structures that are significantly different for different types of nodes. The structure shown in Table 4.2 was developed and used for storing data

of a general node. All the information in this data structure, except the 'AdjBranch', 'AdjNode', 'NewTopo', and 'Topo', is provided by the user. The values of 'AdjBranch', 'AdjNode', 'NewTopo', and 'Topo', are assigned by the software application, and are made available to the user upon request. All data in this structure are invariant except for the 'Status' of the node and 'Voltage' of the node, which is updated continually during on-line operation. The structures of data for busbar and input-output nodes are presented later.

<b>Table 4.2. Node data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>ID</b>	<b>I</b>	<b>Identification number</b>
<b>Name</b>	<b>S</b>	<b>Name (any length)</b>
<b>Type</b>	<b>I</b>	<b>Type of node: common, input, input-output, output</b>
<b>Link1</b>	<b>I</b>	<b>ID number of the link associated with this node</b>
<b>Voltage</b>	<b>F</b>	<b>Current voltage level, kV</b>
<b>Status</b>	<b>CD</b>	<b>Status of the node: de-energized, energized, grounded, healthy, unhealthy</b>
<b>NewTopo</b>	<b>I</b>	<b>Integer representing the topology to be achieved</b>
<b>Topo</b>	<b>I</b>	<b>Integer representing the prevailing topology</b>
<b>AdjBranch</b>	<b>PL</b>	<b>List of adjoining branches</b>
<b>AdjNode</b>	<b>PL</b>	<b>List of adjoining nodes</b>
		<b>Legend: PL = List of integer pointers</b>

#### **4.4.3. Busbars**

Busbars are essentially nodes except for the fact that lines come to converge at the busbars. The system voltages are measured at buses, and these voltages must be within specified limits. The structure shown in Table 4.3 was developed and used for storing data of a busbar. All the information in this data structure, except the 'AdjBranch', 'AdjNode',

'NewTopo' and 'Topo' is provided by the user. The values of 'AdjBranch', 'AdjNode', 'NewTopo', and 'Topo' are assigned by the software application, and are made available to the user upon request. The voltage of a general node is derived by using the voltage measured at the busbars and/or input-output lines. All data in this structure are invariant except for the 'Status' and 'Voltage' of the bus, which are updated continually during on-line operation.

<b>Table 4.3. Busbar data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
ID	I	Identification number
Name	S	Name (any length)
Type	I	Type of node: common, input, input-output, output
Link1	I	ID number of the link on one side of the switch
Voltage	F	Current voltage level, kV
MinV	F	Minimum permissible voltage, kV
MaxV	F	Maximum permissible voltage. KV
Status	CD	Status of the node: de-energized, energized, grounded, healthy, unhealthy, being-repaired, priority
NewTopo	I	Integer representing the topology to be achieved
Topo	I	Integer representing the prevailing topology
AdjBranch	PL	List of adjoining branches
AdjNode	PL	List of adjoining nodes
TransNodes	PL	Transfer nodes, listed according to their priority
Lock	CD	Complex structure defining when it is free to be used

#### **4.4.4. Input-output nodes**

Input-output nodes are different from the general nodes and busbars that they can handle



a specified amount of power. The structure shown in Table 4.4 was developed and used for storing data of an input-output node.

All the information in this data structure, except the 'AdjBranch', 'AdjNode', 'NewTopo', and 'Topo', is provided by the user. The values of 'AdjBranch', 'AdjNode', 'NewTopo', and 'Topo', are assigned by the software application, and are made available to the user upon request. All data in this structure are invariant except for the 'Status', 'Voltage', 'Current' and 'Pf' of the node, which are updated continually during on-line operation.

#### **4.4.5. Transformers**

Transformers can have tap changing features, and rated voltages for the primary, secondary and tertiary windings. They have system buses associated with each side. Keeping these features in mind, the structure shown in Table 4.5 was developed and used for storing data of transformers. All the information in this data structure, except the 'Branch', is provided by the user. The 'Branch' is assigned by the software application, and is made available to the user upon request. All data in this structure are invariant except for the 'Status' and 'TapPos', which are updated continually during on-line operation.

#### **4.4.6. Reactors and capacitors**

Reactors and capacitors are used to keep the voltage of a bus within specified limits. The structure shown in Table 4.6 was developed and used for storing data of reactors and capacitors. All the information in this data structure, except the 'Branch' is provided by the user. The 'Branch', is assigned by the software application, and is made available to the user upon request. All data in this structure are invariant except for the 'Status' and 'Setting' of the reactor, which are updated continually during on-line operation.

#### **4.4.7. Measuring instruments**

The location of measuring instruments, the measured values of the parameters and the status of the instruments are needed for substation control. Considering these factors, the structure shown in Table 4.7 was developed and used for storing data of a measuring

instrument. All the information in this data structure, except the 'Branch', is provided by the user. The 'Branch' is assigned by the software application, and is made available to the user upon request. All data in this structure are invariant except for the 'Status' and 'Value', which are updated continually during on-line operation.

<b>Table 4.4. Input-output node data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>ID</b>	<b>I</b>	Identification number
<b>Name</b>	<b>S</b>	Name (any length)
<b>Type</b>	<b>I</b>	Type of node: common, input, input-output, output
<b>Link1</b>	<b>I</b>	ID number of the link on one side of the switch
<b>SysBus</b>	<b>I</b>	System bus connected to the remote end of the line
<b>Current</b>	<b>F</b>	Actual current, kA
<b>MaxI</b>	<b>F</b>	Maximum permissible current, kA
<b>Voltage</b>	<b>F</b>	Current voltage level, kV
<b>MinV</b>	<b>F</b>	Minimum allowed voltage at the remote end, kV
<b>MaxV</b>	<b>F</b>	Maximum allowed voltage at the remote end, kV
<b>Pf</b>	<b>F</b>	Power factor
<b>MVARating</b>	<b>F</b>	MVA rating
<b>Status</b>	<b>CD</b>	Status of the node: de-energized, energized, grounded, healthy, unhealthy, under-repair, priority
<b>NewTopo</b>	<b>I</b>	Integer representing the topology to be achieved
<b>Topo</b>	<b>I</b>	Integer representing the prevailing topology
<b>AdjBranch</b>	<b>PL</b>	List of adjoining branches
<b>AdjNode</b>	<b>PL</b>	List of adjoining nodes
<b>Buses</b>	<b>PL</b>	Associated buses listed according to their priority
<b>SrcPref</b>	<b>PL</b>	Source list, listed according to preference/priority
<b>Lock</b>	<b>CD</b>	Complex structure defining any special constraints

<b>Table 4.5. Transformer data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>ID</b>	<b>I</b>	Identification number
<b>Name</b>	<b>S</b>	Name (any length)
<b>Type</b>	<b>I</b>	Type of transformer, two-winding, three-winding, auto etc.
<b>MVA</b>	<b>F</b>	MVA rating
<b>TapSideV</b>	<b>F</b>	Tap side or primary voltage rating, kV
<b>ZSideV</b>	<b>F</b>	Z side or secondary voltage rating, kV
<b>TerV</b>	<b>F</b>	Tertiary side voltage rating, kV (if present)
<b>Link1</b>	<b>I</b>	Link on the tap side (or primary side) of the transformer
<b>Link2</b>	<b>I</b>	Link on the impedance (or secondary) side of the transformer
<b>Link3</b>	<b>I</b>	Tertiary side link (normally not present)
<b>SysBus1</b>	<b>I</b>	System bus on the tap (or primary) side
<b>SysBus2</b>	<b>I</b>	System bus on the impedance (or secondary) side
<b>MaxTap</b>	<b>F</b>	Maximum tap setting expressed in p.u., with nominal tap as 1
<b>MinTap</b>	<b>F</b>	Minimum tap setting expressed in p.u., with nominal tap as 1
<b>TapPos</b>	<b>F</b>	Current tap position expressed in p.u., with nominal tap as 1
<b>TapStep</b>	<b>F</b>	Tap step size in p.u. with nominal voltage being as 1
<b>Status</b>	<b>CD</b>	Status of the transformer; healthy, unhealthy, under-maintenance, with-tap, without-tap, controllable, manual, primary-Y, primary- $\Delta$ , secondary-Y, secondary- $\Delta$
<b>Branch</b>	<b>PL</b>	Pointer to the branch(es) in which it belongs
<b>Relations</b>	<b>SL</b>	List of names of associated device separated by commas
<b>Lock</b>	<b>CD</b>	Complex structure defining any special constraints

<b>Table 4.6. Reactor data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>ID</b>	<b>I</b>	<b>Identification number</b>
<b>Name</b>	<b>S</b>	<b>Name (any length)</b>
<b>Type</b>	<b>I</b>	<b>Type of switch: capacitive, reactive, shunt, series</b>
<b>Mva</b>	<b>F</b>	<b>MVA rating</b>
<b>Voltage</b>	<b>F</b>	<b>Voltage rating, kV</b>
<b>Link1</b>	<b>I</b>	<b>Link on one side of the reactor</b>
<b>Link2</b>	<b>I</b>	<b>Link on the other side of the reactor (can be ground node)</b>
<b>Status</b>	<b>CD</b>	<b>Status of the equipment; energized, de-energized, healthy, unhealthy, under-repair, remote-controllable, manual</b>
<b>Branch</b>	<b>P</b>	<b>Pointer to the branch in which it belongs</b>
<b>MaxSetting</b>	<b>F</b>	<b>Maximum setting</b>
<b>MinSetting</b>	<b>F</b>	<b>Minimum setting</b>
<b>Step</b>	<b>F</b>	<b>Steps by which setting changes</b>
<b>Setting</b>	<b>F</b>	<b>Current setting</b>
<b>Relations</b>	<b>SL</b>	<b>List of lines associated with the reactor, separated by commas</b>
<b>Lock</b>	<b>CD</b>	<b>A complex structure, which defines when it is free to operate</b>

#### **4.4.8. Substation**

A substation contains components, branches and nodes. Important elements of substation data are:

- the system bus number represented by the substation,
- the nominal bus voltage,
- file names and directories of relevant data files.

Considering these factors, structure shown in Table 4.8 was developed and used for storing data of a substation. All the information in this data structure are provided by the user.

<b>Table 4.7. Measuring instrument data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>ID</b>	<b>I</b>	<b>Identification number</b>
<b>Name</b>	<b>S</b>	<b>Name (any length)</b>
<b>Type</b>	<b>I</b>	<b>Type of transducer: CT, VT, CVT</b>
<b>Rating</b>	<b>F</b>	<b>Maximum value of measurand that can be measured without saturating the instrument</b>
<b>Value</b>	<b>F</b>	<b>Measured value</b>
<b>Link</b>	<b>I</b>	<b>Link number of the node where the transducer is located</b>
<b>Status</b>	<b>CD</b>	<b>Status of instrumentation; connected, disconnected, healthy, unhealthy, under-repair</b>
<b>Branch</b>	<b>P</b>	<b>Pointer to the branch in which it belongs</b>
<b>Relations</b>	<b>SL</b>	<b>List of associated devices that use this instrument</b>

#### **4.4.9. Power system**

A simple and effective way to check the impact of a switching operation is to conduct a power flow study. The line data and the bus data of the power system must be provided to facilitate these studies. The data can be kept in a data file using specified format. The name of the data file and the directory where the file is kept should be provided to the switching scheme. Once the location of the data is supplied, the switching scheme should perform load-flow calculations to determine if the voltage limits of the buses and the current limits of the lines are violated.

Considering these factors, the structure shown in Table 4.9 was developed and used for storing load flow data of a power system. All the information in this data structure are to be provided by the user except for the status. The status is changed during the on-line operation of the substation.

<b>Table 4.8. Substation data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>Name</b>	<b>S</b>	<b>Substation name</b>
<b>Type</b>	<b>I</b>	<b>Type of substation</b>
<b>SysBus</b>	<b>F</b>	<b>System bus number of the substation</b>
<b>NomV</b>	<b>F</b>	<b>Nominal bus voltage</b>
<b>MinV</b>	<b>F</b>	<b>Minimum permissible bus voltage</b>
<b>MaxV</b>	<b>F</b>	<b>Maximum permissible bus voltage</b>
<b>Status</b>	<b>CD</b>	<b>Status: connected, disconnected, healthy, unhealthy, under-repair</b>
<b>Components</b>	<b>CD</b>	<b>Container to store all the components of the substation</b>
<b>Branches</b>	<b>CD</b>	<b>Container to store all the branches</b>
<b>Nodes</b>	<b>CD</b>	<b>Container to store all the nodes</b>
<b>LFDDir</b>	<b>S</b>	<b>Load-flow data directory</b>
<b>LFIn</b>	<b>S</b>	<b>Load-flow input data file name</b>
<b>LFBusOut</b>	<b>S</b>	<b>Load-flow bus-data output file name</b>
<b>LFLineOut</b>	<b>S</b>	<b>Load-flow line-data output file name</b>
<b>LFProgress</b>	<b>S</b>	<b>Load-flow progress dump file name</b>
<b>SubstData</b>	<b>S</b>	<b>Local substation data input file name</b>
<b>LockExp</b>	<b>S</b>	<b>User defined interlock expression file name</b>

#### **4.4.10. Bus data of the system**

The bus data of the power system must be provided to facilitate load-flow calculations. The data should be kept in a file using specified format. Considering this factor, the structure shown in Table 4.10 was developed and used for storing bus data of a power system. For a load bus, all the data are provided by the user except 'V' and 'VAng'. For a generator bus, all the data are provided by the user except 'QLoad' and 'VAng'. For a generator bus, all the data are provided by the user except 'Pload', 'QLoad' and 'Pgenr'.

<b>Table 4.9. Power system data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
<b>MVABase</b>	<b>F</b>	<b>Base MVA</b>
<b>NoOfBuses</b>	<b>I</b>	<b>Number of buses</b>
<b>BusDataFmt</b>	<b>S</b>	<b>Header to declare the format of bus data</b>
<b>BusData</b>	<b>CDL</b>	<b>Data of all major buses in the system, where each bus has a complex data structure</b>
<b>NoOfBranch</b>	<b>I</b>	<b>Number of branches</b>
<b>BrnDataFmt</b>	<b>S</b>	<b>Header to declare the format of branch data</b>
<b>BranchData</b>	<b>CDL</b>	<b>Data of all major branches in the system, where each branch has a complex data structure</b>

#### **4.4.11. Line Data**

The line data of the power system must be provided to facilitate load-flow calculations. The data should be kept in a file using specified format. The structure shown in Table 4.11 was developed and used for storing line data of a power system. All the information is provided by the user.

### **4.5. Levels of Interlocks**

The interlocks applied to a switch can be processed efficiently if the interlock information is kept in an appropriate format. In this work, all interlock constraints are grouped in four categories to simplify and expedite the processing. These groups are introduced in this section.

#### **4.5.1. Component level interlock**

A switch can be interlocked with several components of the substation. The operation of a switch is prohibited if the components with which it is interlocked are not in specified states.

<b>Table 4.10. System bus data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
Number	I	Bus number
Type	I	Bus type: 0 = Unregulated (load, PQ), 1 = Hold MVAR generation within voltage limits (PQ), 2 = Hold voltage within VAR limits (gen, PV), 3 - Hold voltage and angle (swing, V-Theta)
PLoad	F	Load MW in per unit
QLoad	F	Load MVAR in per unit
PGenr	F	Generation MW in per unit
V	F	Actual voltage in per unit
VAng	F	Actual voltage angle in radians
MinV	F	Minimum voltage limit in per unit
MaxV	F	Maximum voltage limit in per unit
MinP	F	Minimum MW generation limit in per unit
MaxP	F	Maximum MW generation limit in per unit

#### **4.5.2. Substation level interlock**

A substation level interlock prohibits a switching operation from being implemented if the operation leads to pre-specified states. For example, grounding of an energized component is not allowed. Other operations that are monitored by the substation level interlock include the following items:

- removal or addition of a source
- existence of a short circuit path across a switch
- curtailment of a load

Monitoring of these conditions involves the analysis of the topology of the substation and the status of input-output nodes.



<b>Table 4.11: System line data</b>		
<b>Data</b>	<b>Type</b>	<b>Description</b>
Number	I	Line number
TapBus	I	Tap bus number (Bus at the side of the model of a transformer or phase shifter that has the non-unity tap.)
ZBus	I	Z bus number (Bus at the side of the model of a transformer or phase shifter that has the device impedance. )
Type	I	Type:0 = Transmission line, 1 = Fixed tap, 2 = Variable tap for voltage control (TCUL, LTC), 3 = Variable tap (turns ratio) for MVAR control, 4 = Variable phase angle for MW control (phase shifter)
R	F	Branch resistance R, per unit
X	F	Branch reactance X, per unit
Bby2	F	Line charging B/2, per unit
TurnsRatio	F	Transformer final turns ratio
AppPower	F	Line MVA rating in per unit

#### **4.5.3. System level interlocks**

A system level interlock prohibits a switching operation from being implemented if it is likely to cause overloading of lines, or under/over voltage of busbars. A switching operation at a substation can redistribute the current and voltage profiles in the entire system and can result in undesirable operating conditions. The load flow studies are carried out to find the likelihood of overloads and out-of-limit voltages.

#### **4.5.4. Manual interlocks**

In addition to interlocks described in Section 4.5.1-4.5.3, provisions should be made to manually block and release switches. A switch must be blocked when it is known that it

is not likely to operate properly. The blocking does not allow the switch to be opened or closed until it is released. The manual blocking of a switch is also necessary when personnel are to do repair work or maintenance of the switch.

#### **4.6. Summary**

The constraints for operating switches and other equipment in a substation have been outlined in this chapter. The constraints, which must be examined when a sequence of operations is to be performed, are identified. The data of various components used in preparing switching sequences and interlocking have been identified. Some examples of switching procedures have been described.

## **5. SWITCHING TECHNIQUE - MAJOR COMPONENTS**

The operating constraints, data requirements and types of switching procedures involved in a switching scheme are reviewed in Chapter 4. A technique for developing switching schemes that incorporate those features is presented in this chapter. The technique is capable of addressing specific requirements of substations. The technique is geared for providing a tool that would allow substation engineers to develop switching schemes for substations of different configurations.

Many software components play key roles in the development of a switching scheme. The major components are

- a protocol for handling the status of the circuit components,
- a technique to process the topology of a substation,
- a technique to carry power flow studies and estimate the system states,
- a technique to determine alternate paths of power flow,
- a protocol for handling the interlocks, and
- a switching sequence generating technique.

### **5.1. Status Handling**

Parameters of the components of a substation change during its operation. These parameters should be continuously monitored. If integer variables were assigned to represent each

status parameter of a component, substantial disk and memory storage would be required. In addition, processors will take more time if the parameters were to be compared individually with set values. It would be advantageous if several parameters were simultaneously compared with their set values.

The management of status information would be convenient if the status of a component were stored in a specially packaged variable such that it would be possible to compare these status variables with the set values in one pass. A protocol was, therefore, developed for this purpose. A syntax was formulated for a construction and comparison of status variables based on this protocol, and exchange of status information with other components.

#### **5.1.1. Uniform status storage and processing protocol**

The protocol, *uniform status storage and processing (USSP) protocol*, was developed and used to store the status information of all the components in a substation, update them on-line, and compare them with set values. This protocol has seven major components.

1. A status management program, *status management unit* (SMU), computes, compares and stores the status of a component. The status of each component is handled by its SMU.
2. Software interface functions facilitate the communication of status information between the SMUs and other modules.
3. Integers that are multiples of 32 bits provide space for allocating bits to represent the status parameters of a component in a SMU. The meanings of bits, however, can be different in the status variables used for different components.
4. The floating point parameters, such as line currents, bus voltages and power factors, are stored in separate variables. These parameters are then compared with their set limits and the results of the comparisons are included in the SMU. A provision was made to handle multi-valued states

such as unhealthy, marginal and healthy.

5. Some key information, such as grounded, live and dead status of an equipment, is acquired from the topology analysis and stored as boolean status fields in the SMU. The network topology continually changes as the switches open and close. The topology is not represented by status variables. It is stored separately and is dealt by a separate program module.
6. A set of standard keywords were used to refer to the state of a component. The keywords and their meanings are listed in Table 5.1. All the status parameters listed in the table are not relevant for all components. Only those status parameters that are relevant to a component are tracked by its SMU.
7. The values of all status parameters were made available to other components upon request.

<b>Table 5.1. The status keywords and their meanings</b>		
<b>Keyword</b>	<b>Applicable To</b>	<b>Meaning of the keyword</b>
<b>MAKEI</b>	Switch	Makes current if closed.
<b>NOMAKEI</b>	Switch	Does not make current if closed.
<b>BREAKI</b>	Switch	Breaks current if opened.
<b>NOBREAKI</b>	Switch	Does not break current if opened.
<b>FREE</b>	Switch	Free meaning that it is not stuck, but it is not a green signal to operate.
<b>STUCK</b>	Switch	The switch is stuck, cannot be operated.
<b>MANLOCK</b>	Switch	Manual lock means that the operator has specifically instructed the switch to not operate until the operator takes this instruction back, even if all other conditions are conducive for operation.

NOMANLOCK	Switch	The manual lock is released. <b>Please note that this is different from the interlocks placed through the interlock expressions to be discussed later in this chapter.</b>
ON	Switch, branch, node	The switch or a branch is in a closed state. The node is energized.
OFF	Switch, branch, node	Switch, branch: The component is in open state. Others: The component is de-energized.
LOCKED	Switch, Branch, node	Interlock rules say it should not operate (or energized/de-energized).
UNLOCKED	Switch, Branch, node	Its operation is not blocked by the interlock rules.
INCLUDE	Switch, Branch, node	Include the component in the power flow paths.
EXCLUDE	Switch, Branch, node	Exclude the component from power flow paths.
RIDDEFAULT	Switch, branch, node	Override the default interlocks
USEDEFAULT	Switch, branch, node	Do not override the default interlock expression
BYPASS	Switch, branch, node	Bypass the component from the power flow paths
NOBYPASS	Switch, branch, node	Do not bypass the component from the power flow paths
NORMI	I/O node	Normal current, which means the line is normally loaded.
OVERI	I/O node	Over current, which means the line is over-loaded.
EXTLIVE	I/O node	External circuit that is connected to this node is live.

EXTDEAD	I/O node	External circuit that is connected to the node is not live.
REMOTEISOL	I/O node	The input-output line is isolated from the remote end.
NOREMOTEISOL	I/O node	The input-output line is not isolated from the remote end.
LOADED	I/O node	Load is connected to this output node.
OFFLOADED	I/O node	Load is not connected to this output node.
TRANSIT	Node	Changes from energized to de-energised state or vice versa.
NOTRANSIT	Node	Does not change from energized to de-energised state or vice versa.
NORMV	Node	The voltage is within the specified limits.
OFFLIMITV	Node	The voltage is outside the specified limits.
ENABLED	Any	It is free, functional, unlocked, healthy, not-disabled and can be operated or used if the substation and the system have no objections.
DISABLED	Any	It can not be operated or used due to one of many possible reasons.
HEALTHY	Any	The component is problem free.
UNHEALTHY	Any	A problem has been encountered on the component and further attempts must not be made to operate/use it before the problem is resolved.
REPAIR	Any	The component is under repair. A crew is working or preparing to work on it.
FUNCTIONAL	Any	The component is not under repair.
ISOLATED	Any	All the isolators surrounding this component have been opened and locked.

NOTISOLATED	Any	The component is connected or can potentially be connected to the system.
GROUNDDED	Any	The component is connected to the ground node.
NOTGROUNDDED	Any	The component is not connected to the ground node.
SCPATH	Any	A short circuit path exists across the component.
NOSCPATH	Any	No short circuit path exists across the component.
NORMGENRN	System	Level of generation is adequate.
LOWGENRN	System	Level of generation is inadequate.

### 5.1.2. Syntax for specifying status

A human interface for specifying and testing the status parameters of components was prepared. This interface converts the text-based instructions to the binary form for use by SMUs.

The SMU of a component stores and maintains the required number of status parameters. A status is defined by the name of the object followed by its status parameters placed between parentheses. The status parameters are separated by blank spaces. The syntax of a status definition is:

**NameOfObject(param1 param2 param3 .... paramn)**

Three examples of the syntax are as follows.

**Breaker1(ON ENABLED UNLOCKED)**

**Breaker2(OFF)**

**Busbar1(ISOLATED GROUNDED)**

In these examples Breaker1, Breaker2 and Busbar1 are the names of the objects whose



status are stated. ON, ENABLED, UNLOCKED, OFF, ISOLATED and GROUNDED are the status parameters. Breaker1 must be ON, ENABLED and UNLOCKED for TRUE outcome of the first statement.

### **5.1.3. Handling of on-line status**

Global handling of status makes a scheme vulnerable to data errors. It becomes difficult to determine the culprit code or device that corrupts the data if many devices and program units are responsible for maintaining the same data. It was, therefore, decided that the storage and maintenance of status be handled locally by the software model of the device. This was done to make the information management more robust and to reduce the impacts of malfunctions of status units.

Every time a switch operates, its SMU refreshes its status variables to reflect the new state. The SMUs of other components also refresh their status variables. Every time the topology of the substation is processed, some states, such as live, dead, and grounded, are passed on by the topology processor to the SMUs of affected components. Each SMU updates the status variables. Every time current and voltage values are updated, the comparisons are made with set limits and the results of the comparisons are sent to SMUs of affected components. The SMUs update the status variables.

Interface functions were defined for reading and comparing status parameters of a component. The following interfaces must be supported by all SMUs of components:

```
<return_value1> GetStatus( )  
<return_value2> GetStatusString( )  
<return_value2> GetKeyString(<keys> )  
<return_value3> CheckAll(<test_status> )  
<return_value3> CheckSelective(<test_status>, <key>)
```

where,

<return\_value1> is the copy of the entire packed structure of the status parameters, which is referred to as Status.

**<return\_value2>** is a string, that explains the status of a component in well formed test strings.

**<return\_value3>** is an integer, which is 1 if the test result is TRUE and 0 if FALSE.

**<test\_status>** is a Status of some other component to be compared with the Status of present component to find equality of all parameters.

**<key>** is a Status that tells which bits of the component are to be compared with the **<test\_status>**. The bits to be compared are set to 1 and those not to be compared are set to 0.

**GetStatus** returns the Status of the component

**GetStatusString** returns a text string explaining the Status of the component

**GetStatusString** returns a text string describing the key

**CheckAll** compares all the parameters of the Status of the component with the input Status. Returns 1 if both Status are equal.

**CheckSelective** compares selected parameters. Only those parameters for whom the corresponding bits in **<key>** are set to 1 are compared. Returns 1 if selected fields in both Status are equal.

The SMUs should also support the all those applicable to the component watched by the SMU among the following interface functions, which will return 1 if the result was TRUE and 0 if FALSE. The meaning of these interface functions are the same as their corresponding labels as defined in Table 5.1.

**MakeI(), BreakI(), Free(), ManLocked(), On(), Locked(), RidDefault(), Bypass(), NormI(), ExtLive(), RemoteIsol(), Loaded(), Transit(), NormV(), OffLimitV(), Enabled(), Healthy(), Repair(), Isolated(), Grounded(), SCPath(), NormGenrn()**

#### **5.1.4. Handling of power flow results**

The currents that would flow in the lines and the voltages that would be experienced at the

system buses after a switching operation are calculated to check that the operating limits of the components are not violated. These checks are performed by first calculating the power flows and then comparing the current in each line with its continuous current rating and comparing the bus voltage magnitude of each bus with the maximum and minimum permissible voltage specified for the bus. The technique that was selected for power flow is the decoupled power flow technique [37]. This technique is described in appendix A.

## **5.2. Substation Topology**

Topology determination is a process of scanning the network to find which circuit elements are electrically connected. In this application, the components of interest are transmission lines, busbars, generators, transformers, and loads. The determination of network topology is important not only for interlocking and sequence switching, but also for state estimation, security analysis, power system protection and other applications.

The impedance elements within the substation, such as transformers, are ignored while determining the topology of the network. Starting from a particular node, the network can be searched to determine if it is connected to other nodes. Most of the topology analysis algorithms are modifications of well known network search techniques, such as the breadth-first-search and depth-first-search techniques. The works of Dy Liacco and Kraynak [ 30], Dy Liacco, Ramaro and Weiner [31], Sason et al. [32], Couch and Morrison [33], Sullivan, Reichert and Saly [34], Bertran and Corbella [35], Prais and Bose [36] describe topology determination techniques for power system networks. These papers deal with the topology determination of a power network limited to the interconnection of lines and transformers.

A technique, which was developed by the author and used in this research, for determining the topology of a substation is presented in this section.

### **5.2.1. Topology representation**

The following procedure was developed for representing the topology of a substation and for exchanging the topology information between modules, such as SMUs.

1. A substation is considered to be made up of nodes and branches. A branch is made up of series connected elements. A node is a terminal point of a branch.
2. The topology information of the substation is stored in terms of the connectivity of nodes.
3. All the nodes of the substation are mapped with two sets of integers. The first set of integers, *current-topology-set*, represents the actual topology of the substation. The second set, *next-topology-set*, represents the topology of the substation, after the required switching operations are completed. The values in these sets are called *topology-indicators* of the corresponding nodes.
4. In the *current-topology-set*, topology-indicators of two nodes are equal if the nodes are electrically connected. They are different if the nodes are electrically disconnected. Similarly, in the *next-topology-set*, *topology-indicators* corresponding to two nodes are equal if the nodes are going to be electrically connected after the implementation of a switching sequence. They are different if the nodes are not going to be electrically connected.
5. Two integer fields, “Topo” and “NewTopo” were allocated in the data fields of nodes (Table 4.2-4.4) to store the *topology-indicators* belonging to the *current-topology-set* and the *next-topology-set*.

### 5.2.2. Topology Determination

A technique for determining the topology of a substation is explained in this section by using the substation shown in Figure 5.1 as an example. In this substation, the six isolators are 1, 2, 3, 4, 5, and 6 and the three circuit breakers B1, B2 and B3. The links that form the terminals of switches are identified as **1, 2, 3, 4, 5, 6, 7, and 8**. It is essential that each link be uniquely identified. More than one switch can be connected to a common link and, therefore, share a link number. For example, switches 2, 4 and 6 share a common link **4**.

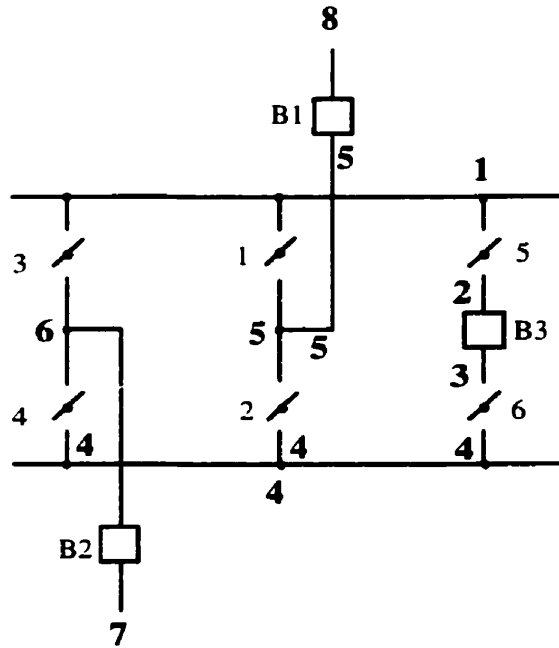
Switches 5, B3 and 6 of Figure 5.1 form one branch. The links **1, 4, 5, 6, 7, 8** also identify the nodes of this substation. The identification number of a node is shared with one or more

links. In this thesis, the topology is expressed in terms of the connectivity of nodes.

When all switches of a substation are open, all nodes are isolated from one another and the topology-indicator for each node is equal to its identification number. The topology of the substation of Figure 5.1 can be represented by the following *current-topology-set*.

**[1, 4, 5, 6, 7, 8, 0]**

Note that links 2 and 3 are not considered as they are not nodes. The node 0 represents the ground node.



**Figure 5.1.** A segment of a substation circuit

If the line identified as node 8 were to be connected to the bus identified as node 1 by closing switches 1 and B1, the topology-sets would be as follows.

*current-topology-set:* **[1, 4, 5, 6, 7, 8, 0]**

*next-topology-set:* **[1, 4, 1, 6, 7, 1, 0]**

The closing of switches 1 and B1 would connect node 8 to node 5, and node 5 to node 1.

Nodes 8 and 5 would now be electrically identical to node 1. The next-topology-set is obtained by replacing 5 and 8 with 1.

The topology processor stores these topology sets in the data structures of the nodes. The topology indicators of the nodes 1, 5 and 8 are shown in Table 5.2. Two data fields store the topology-indicators, the first (Topo) stores the actual topology and the second (NextTopo) stores the topology to be attained.

The topology processor also sends the required status information to the SMUs of affected nodes and switches. Node 8 was originally in the ON state. If nodes 1 and 5 were connected to node 8, they would be energized. These changes are reflected by the entries in ON and TRANSIT status variables. This switching operation does not make any currents to flow through the nodes, which is reflected by the entries in MAKEI.

<b>Table 5.2. Data before energizing Bus 1</b>										
	Node 1		Node 5		Node 8		Switch 1		Switch B1	
	This	Next	This	Next	This	Next	This	Next	This	Next
Topo	1	1	5	1	8	1	N/A	N/A	N/A	N/A
ON	F	T	F	T	T	T	F	T	F	T
MAKEI	N/A	N/A	N/A	N/A	N/A	N/A	F	F	F	F
BREAKI	N/A	N/A	N/A	N/A	N/A	N/A	F	F	F	F
TRANSIT	F	T	F	T	F	F	N/A	N/A	N/A	N/A
GND	F	F	F	F	F	F				
T = TRUE, F = FALSE, N/A = not applicable, GND = GROUNDED, This = prevailing state, Next = state-to-be, BLANK = Not processed by topology processor										

Now, if switches 1 and B1 are closed, the *current-topology-set* would change to the following form:

*current-topology-set: [1, 4, 1, 6, 7, 1, 0]*

These changes are reflected in the topology-indicators and status variables as shown in Table 5.3. The table shows that nodes 1, 5 and 8 are all ON. Variable TRANSIT indicates that the ON state of 1 and 5 was changed due to the switching operation.

At this stage, if the line represented by link 7 were to be connected to the bus identified as node 1 by closing switches 3 and B2, the topology-sets would be as follows.

*current-topology-set:* [1, 4, 1, 6, 7, 1, 0]

*next-topology-set:* [1, 4, 1, 1, 1, 1, 0]

Table 5.3. Data after energizing Bus 1										
	Node 1		Node 5		Node 8		Switch 1		Switch B1	
	This	Next	This	Next	This	Next	This	Next	This	Next
Topo	1	1	1	1	1	1	N/A	N/A	N/A	N/A
ON	T		T		T		T		T	
MAKEI	N/A	N/A	N/A	N/A	N/A	N/A	F	F	F	F
BREAKI	N/A	N/A	N/A	N/A	N/A	N/A	F	F	F	F
TRANSIT	T		T		F		N/A	N/A	N/A	N/A
GND	F		F		F					

The status variables and topology indicators of nodes 1, 6 and 7, and the status variables of switches 3 and B2 assume the values shown in Table 5.4.

Note that MAKEI variable indicates that current will be made due to the operation of these switches. The making of current is apparent as previously de-energized line connected to node 7 is now energized. After the switches 3 and B2 are closed, the *current-topology-set* would change to the following form:

*current-topology-set:* [1, 4, 1, 1, 1, 1, 0]

<b>Table 5.4. Data before energizing node 7</b>										
	Node 1		Node 6		Node 7		Switch 3		Switch B2	
	This	Next	This	Next	This	Next	This	Next	This	Next
Topo	1	1	5	1	7	1	N/A	N/A	N/A	N/A
ON	T	T	F	T	F	T	F	T	F	T
MAKEI	N/A	N/A	N/A	N/A	N/A	N/A	F	F	F	T
BREAKI	N/A	N/A	N/A	N/A	N/A	N/A	F	F	F	F
TRANSIT	F	F	F	T	F	T	N/A	N/A	N/A	N/A
GND	F	F	F	F	F	F				

The status variables and topology-indicators assume the data as indicated in Table 5.5.

<b>Table 5.5. Data after energizing node 7</b>										
	Node 1		Node 6		Node 7		Switch 3		Switch B2	
	This	Next	This	Next	This	Next	This	Next	This	Next
Topo	1	1	5	1	7	1	N/A	N/A	N/A	N/A
ON	T		T		T		T		T	
MAKEI	N/A	N/A	N/A	N/A	N/A	N/A	F		T	
BREAKI	N/A	N/A	N/A	N/A	N/A	N/A	F		F	
TRANSIT	F		T		T		N/A	N/A	N/A	N/A
GND	F	F	F	F	F	F				

If the two busbars (nodes 1 and 4) were to be connected by closing switches 5, B3 and 6, all the nodes in the circuit become electrically connected and the topology sets would be as follows.

*current-topology-set:* [1, 4, 1, 1, 1, 1, 0]

*next-topology-set:* [1, 1, 1, 1, 1, 1, 0]



The status variables and topology indicators of nodes 1, 5 and 4, and the status variables of switches 2 and 4 assume the values shown in Table 5.6. The ON and TRANSIT variables indicate that node 4 will be energized due to the closing of switches 5, B3 and 6. The SCPATH variable indicates that there will be a short-circuit path across switches 2 and 4. The existence of a short-circuit path across switch 2 is apparent from the fact that the nodes 5 and 4 are electrically connected while switch 2 is still open. Also, node 6 is connected to node 1 that is connected to node 4. The bus coupler circuit, in this manner, has provided a short circuit path across isolator 4.

Table 5.6. Data before energizing node 4										
	Node 1		Node 5		Node 4		Switch 2		Switch 4	
	This	Next	This	Next	This	Next	This	Next	This	Next
Topo	1	1	1	1	4	1	N/A	N/A	N/A	N/A
ON	T	T	T	T	F	T				
MAKEI	N/A	N/A	N/A	N/A	N/A	N/A				
BREAKI	N/A	N/A	N/A	N/A	N/A	N/A				
SCPATH	N/A	N/A	N/A	N/A	N/A	N/A	F	T	F	T
TRANSIT	F	F	F	F	F	T	N/A	N/A	N/A	N/A
GND	F	F	F	F	F	F				

After the switches 5, B3 and 6 are closed, as shown in Figure 5.2, the *current-topology-set* would change to the following form.

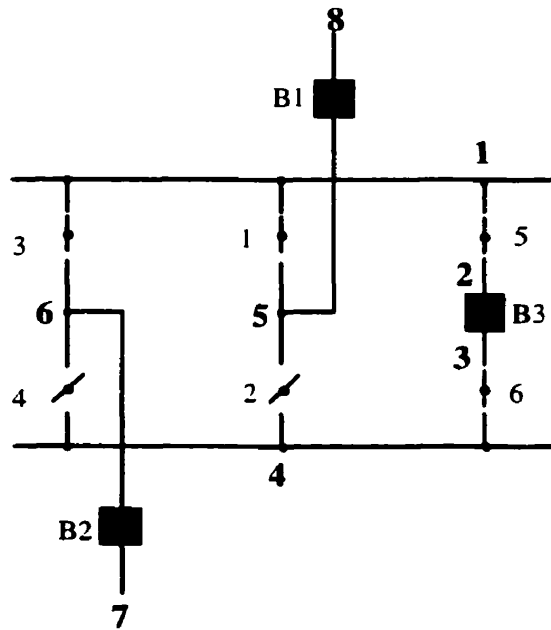
*current-topology-ser:* [1, 1, 1, 1, 1, 1, 0]

Other conditions, such as grounding of a node and breaking of current by a switch, are also determined by reading the status variables of the components in a similar fashion.

The procedure of updating the topology of a substation can be summarized in the following two steps.

1. The *next-topology-set* is prepared when a switch is to be opened or closed.
2. The *current-topology-set* is updated as soon as the switching operation is completed.

The examples presented in this section have demonstrated the procedure of processing topology and keeping the topology information in the data structures of nodes. The examples have also demonstrated the role that a topology processor plays in determining and updating the values of various status variables.



**Figure 5.2.** Illustration of a shunt path.

### 5.3. Searching Power Flow Paths

Alternative power flow paths between two nodes of a substation are determined by tracing the network of the substation. While searching for the paths, a number of limiting factors are considered. An approach developed and used in this thesis for the search of power flow

paths is explained in this section by using the substation shown in Figure 5.1 as an example.

A list of adjoining nodes is prepared for each node. For example, the adjoining node of node 8 is node 5 and the adjoining nodes of node 5 are nodes 1, 4 and 8. The adjoining nodes of the nodes in the substation determined in this manner are listed in Table 5.7.

<b>Table 5.7. List of adjoining nodes</b>	
<b>Node</b>	<b>Adjoining nodes</b>
<b>1</b>	<b>4, 5, 6</b>
<b>4</b>	<b>1, 5, 6</b>
<b>5</b>	<b>1, 4, 8</b>
<b>6</b>	<b>1, 4, 7</b>
<b>7</b>	<b>6</b>
<b>8</b>	<b>5</b>

If alternate paths between node 8 and node 1 were to be determined, node 8 is declared as the start-node and node 1 is declared as the destination-node. Adjoining nodes of node 8 are found. In this case, the adjoining node is node 5. Now, the adjoining nodes of node 5 except node 8 are identified. These nodes are 1 and 4. Since node 1 is a destination-node, one possible path has been found to be from nodes 8 to node 5, and from node 5 to node 1.

The search of other paths continues from node 4, whose adjoining nodes are 1, 5 and 6. Node 1 is destination-node, and therefore a new path has been found to be from node 8 to node 5, node 5 to node 4, and node 4 to node 1. Another traced path would be 8-5-4-5. This path is rejected because a path should not traverse a node more than once.

The search now reverts to node 6. The adjoining nodes of node 6 are nodes 1, 4, and 7. The path from node 6 to node 1 cannot be used because the path from node 4 to node 6 has been used. This restriction is based on the interlocking requirements between switches 3 and 4.

Node 7 is rejected as an end of the substation is reached.

The paths that can be established between nodes 8 and 1 are, therefore,

**8, 5, 1**

and **8, 5, 4, 1.**

The list of adjoining nodes are updated continuously by removing the nodes that can not be used in the power flow path and adding them to the list as soon as they can be used. For example if the status variable LOCKED, STUCK, UNHEALTHY, REPAIR, or MANLOCK of switch 2 of Figure 5.1 were TRUE, the list of adjoining nodes of nodes 5 and 4 would be modified to be as shown in Table 5.8. The lists of adjoining nodes of nodes 4 and 5 would be reverted back to those shown in Table 5.7 when switch 2 becomes useable.

<b>Table 5.8. Modified list of adjoining nodes</b>	
<b>Node</b>	<b>Adjoining nodes</b>
<b>4</b>	<b>1, 6</b>
<b>5</b>	<b>1, 8</b>

These examples have demonstrated the procedure of searching power flow paths.

## **5.4. Determining Switching Sequences**

A switching sequence is composed of basic operations such as

- connecting a node to another node and
- disconnecting a node from another node.

The following algorithm was developed and used for determining the switching sequences.

### **5.4.1 Connecting two nodes**

The following procedure was developed for determining the switching sequences for

connecting two nodes.

1. Determine all alternate paths between the two nodes.
2. Select the shortest path.
3. In the selected path, identify all branches that contain isolators only. Consider that all the isolators have been closed. Now check the following.
  - (i) Does the closing of the isolators cause the currents to flow?
  - (ii) Does the closing of the isolators ground a live node?
  - (iii) If the result of the test 3(i) and 3(ii) is FALSE, proceed to Step 4.
    - (a) Reject the path that was being tested.
    - (b) Select the next shortest path for test. Revert to Step 3 and continue testing the newly selected path.
4. Consider that all the isolators identified in Step 3 remain closed. In the selected path, identify all branches that contain circuit breakers. Consider that all the circuit breakers and isolators adjoining them are closed. Now check the following.
  - (i) Does the closing of the switches ground a live node?
  - (ii) If the result of test 4(i) is FALSE, proceed to Step 4(iii).
    - (a) Reject the path that was being tested.
    - (b) Select the next shortest path for test. Revert to Step 3 and continue testing the newly selected path.
  - (iii) Does the closing of the switches cause the currents to flow?
  - (iv) If the result of test 4(iii) is FALSE, proceed to Step 4(v).
    - a) Check from the load-flow solution if any operating limit is violated.
    - b) If no limit is violated, proceed to Step 4(v).
      - Reject the path that was being tested.
      - Select the next shortest path for test. Revert to Step 3 and

continue testing the newly selected path.

- (v) Check if two circuits with unsynchronized voltages are being connected.
- (vi) If the result of Step 4(v) is FALSE, proceed to Step 5.
  - Print the instruction to synchronize the voltage.

5. Make a list of all branches in the selected path using the following steps.

- (i) Include in the list the branches identified in Step 3. Arrange them sequentially from source to load.
- (ii) Include in the list the branches identified in Step 4. Arrange them sequentially from source to load.

The list of branches prepared in Step 5 represents the sequence in which the operations must be performed. The list is in such a form that the two nodes can be connected by following the instructions in the list sequentially from the top.

#### **5.4.2 Disconnecting two nodes**

The following procedure was developed for determining the switching sequences for disconnecting two nodes.

1. Determine all alternate paths between the two nodes. Sort the paths by their lengths so that the shortest path appears at the top of the list.
2. Eliminate the paths that are already open.
3. Prepare an empty list of branches named solution-list.
4. Select the path that is at the top of the list.
5. Find if the path contains a branch that is already included in the solution-list. If yes, eliminate the path from the list and revert to Step 4.
6. Identify all branches that contain circuit breakers. Consider that all the circuit breakers and isolators adjoining them have been opened. Now, check the following.
  - (i) Does the opening of the switches break the flow of currents?

- (ii) If the result of test 6(i) is FALSE, proceed to Step 6.(iii).
  - (a) Test from the load-flow solution if the operating limit of any equipment is violated.
  - (b) If an operating limit is not violated, proceed to Step 7.
    - Reject the path that was being tested. Print a message indicating that the operating limits will be violated.
    - Terminate the process.
- 7. Consider that the switches identified in Step 6 remain open. Identify all branches that contain isolators only. Consider that all the isolators have been opened. Now, check the following.
  - (i) Does the opening of the isolators interrupt the flow of current?
  - (ii) If the flow of currents is not interrupted, proceed to Step 8.
    - (a) Reject the path that was being tested.
    - (b) If the path was rejected for the first time, remove the path from the top of the list and insert it at the bottom. Revert to Step 4. Otherwise, print a message and terminate the process of finding the switching sequence.
- 8. Make a list of all branches in the path. Prepare the list by using the following steps.
  - (i) Include in a list the branches identified in Step 6. Arrange them sequentially from load to source.
  - (ii) Include in a list the branches identified in Step 7. Arrange them sequentially from load to source.
  - (iii) Add the lists into the solution-list.
- 9. Remove the tested path from the top of the list.
- 10. If the list of paths is not empty, proceed to Step 4. Otherwise, the solution-list represents the desired sequence for disconnecting the nodes.

Follow instructions contained in the list sequentially from the top when implementing the solution.

## **5.5. Generalized Sequence Switching Technique**

Switching procedures were classified into two groups, simple and complex. Simple operations, such as connecting nodes and disconnecting nodes, have been described in Section 5.6. Complex switching operations are discussed in this section.

Complex switching tasks were systematically divided into smaller and generalized tasks. An example of a complex task is to transfer lines from Bus1 to Bus2 in the substation shown in Figure 5.3. This task was performed by sequentially performing the following tasks.

- Determine the lines that are connected to Bus1 (Source1, Source2, Load1, Load2).
- Connect Bus2 with Bus1 (close 1, 2 and B5).
- Connect the identified lines that were previously connected to Bus1 to Bus 2 (close 4, 6, 8 and 10).
- Disconnect the lines from Bus 1 (open 3, 5, 7 and 9).
- Disconnect Bus 1 from Bus 2 (open B5, 1, and 2).

The following guidelines were formulated to facilitate the handling of complex tasks.

1. A dedicated *command handler unit* (CHU) was provided for each component. The CHU accepts user commands, processes them, makes decisions and takes actions.
2. The CHU of a component communicates with the CHU of other components.
3. A set of keywords were assigned as command names. The names of some important commands and their meanings are listed in Table 5.9. A CHU can accept the commands and convert them to appropriate machine code.
4. Text-strings containing the name of the component followed by a period and then



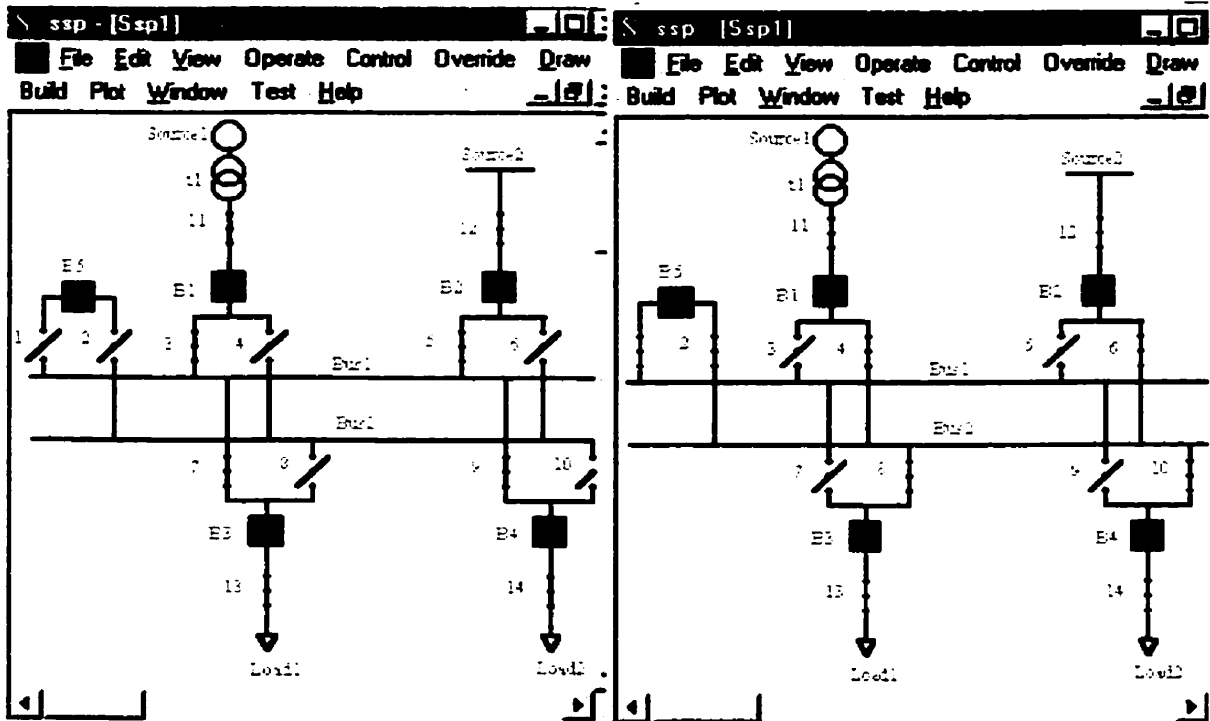
the name of a command were used to form Command-Strings as follows.

### **ComponentName.CommandName**

For example, the Command-String for opening CB1 is as follows.

### **CB1.Open**

5. CHU issues only one command at a time. If several commands are required to be issued, they are issued in a sequence.



**Figure 5.3.** A bus-transfer process under progress.

6. A complex command is formed by preparing a list of already defined commands.
7. A complex command contains the name of the component and a Command-String-List, separated by a colon as is shown in the following statement.

### **ComponentName : Command-String-List**

8. The Command-String-List contains Command-Strings separated by blank spaces

as is shown in the following statement.

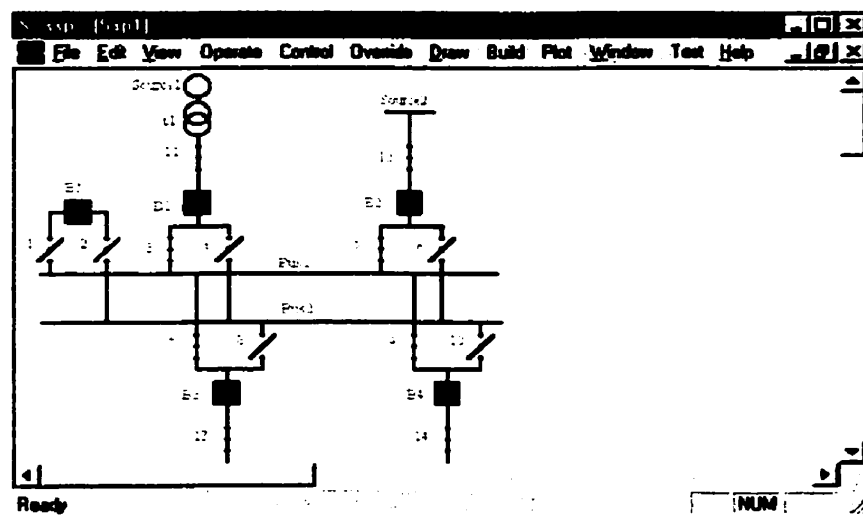
**Owner1.Command1 Owner2.Command2 ..... OwnerN.CommandN**

An example of a complex switching command formed to isolate circuit breaker B1 shown in Figure 5.4 is as follows.

**B1 : Source1.DisconnectFromBus 11.Open 3.Open 4.Open**

<b>Table 5.9. Command names</b>	
<b>Name</b>	<b>Description</b>
Inoperable	Declare the component as inoperable.
Operable	Declare the component as operable. An interlock constraint could forbid the operation.
MLock	Manually lock the component overriding other constraints.
MUnlock	Remove the manual lock. The component could be locked by other interlock constraints.
Topology	Determine the topology of the network.
EnergizedNodes	Determine the nodes that are energized.
IOTransition	Determine if a new line has been connected or if a previously connected line has been disconnected.
FindPaths	Find paths between a node and a number of other nodes.
LoadFlow	Perform load-flow analysis.
Close	Close a switch or branch.
Open	Open a switch or a branch.
OpenAll	Open all switches in the branch, or disconnect all lines from the busbars in the substation.
CloseAll	Close all switches in the branch, or connect all lines to the busbar if feasible.
Isolate	Isolate the component. There must be an open isolator along each path radiating out from the component.

Table 5.9. Command names	
Name	Description
Energize	Connect the node to an active source.
ConnectToBus	Connect to an available bus.
ConnectToMBus	Connect to the designated (main) bus.
ConnectTo(comp)	Connect to the node whose name is given as an argument.
Disconnect	Disconnect the node from the rest of the substation.
DisconnectFromBus	Disconnect the node from the buses.
DisconnectFrom(comp)	Disconnect from the node whose name is given as an argument.
TransferToAny	Transfer the line connections from this bus to an alternate bus.
TransferTo(BusName)	Transfer the line connections from this bus to the bus whose name is supplied as the argument.



**Figure 5.4.** A section of a double-bus substation.

Each class of components accepts a different set of commands. The following examples indicate the difference in commands developed for Switch, Node, Branch and Substation.

1. Switch: Close, Open, Isolate, Repair
2. Node: ConnectToBus, ConnectToSource, DisconnectFromNode, Isolate, Energize, Deenergize, ConnectToGround, Repair
3. Branch: Close, Open, OpenAll, Isolate
4. Substation: CloseAll, OpenAll, Restore

The guidelines and examples presented in this section have demonstrated the method of developing complex switching sequences by arranging already defined commands.

## 5.6. Generalized Interlocking Technique

All the components whose operations are interlocked with other components were modelled. The modelling followed a procedure in storing, managing and communicating the information. The protocol was developed and used to facilitate the design and enhancement of interlocking schemes. The developed procedure can include new classes of components with minimal effort. The procedure involves the following steps.

1. An interlock management program, *interlock management unit* (IMU), stores the interlock constraints, interprets them, and continuously updates the LOCKED/ UNLOCKED status variable residing in the SMU of the component.
2. The IMUs accept the interlock constraints in the form of logical expressions of modules called *lock units* (LU). These modules store the names of the components, the status-variables to be compared, and the desired values of those variables.
3. For use by the user-interface, a *lock unit* (LU) is defined by the name of an object followed by its desired status parameters placed between parentheses. The status parameters are separated by blank spaces. Only the keywords specified in Section 5.1 and applicable to the component are acceptable. The result of the evaluation of a LU is always TRUE or FALSE, and the syntax of a LU definition is as follows.

**NameOfObject(param1 param2 param3 .... paramn)**

4. The status fields in a LU must not conflict with each other. For example, ON OFF are two conflicting status values and cannot both be included in the same LU.
5. An IMU stores interlock logic in the form of a *Lock String* (LS). A LS is an expression made up of LUs concatenated by AND (&) and OR (!) operators, negated by NOT (!) operator and the order of evaluation altered by parentheses. An evaluation of a LS returns TRUE or FALSE, where the return of TRUE means that the component should be locked.
6. A LS is defined by the name of the object followed by a colon, which is followed by an expression composed of LUs separated by logical operators.

**NameOfObject: Logical-Expression-of-LUs**

7. An interpreter is needed by the IMUs to translate the text-based LSs into binary code and vice versa.
8. Besides a *Lock String*(LS), an IMU can optionally store a *Default Lock String* (DLS) and *Override Lock String* (OLS). The formats of the DLS and OLS are the same as that used to define a LS. The outcome of an interlock is evaluated as TRUE only when the DLS or the LS returns TRUE and the OLS returns FALSE. A LS, DLS or OLS is not evaluated if it is empty.

A few examples of interlock expressions for switching operations in the substation shown in Figure 5.4 are now presented.

***Example 1:***

The following is an example of a valid Lock Unit.

**B1(ON ENABLED UNLOCKED)**

In this LU, B1 is the name of the object, and **ON**, **ENABLED** and **UNLOCKED** are the

status flags whose values must be logical TRUE for the LU to be evaluated as TRUE.

**B1(ON OFF ENABLED)** is an invalid lock unit as it has two conflicting status values. A component cannot be ON and OFF at the same time. Lock unit, **B1(ON ENABLED)**, on the other hand, is valid because it does not have conflicting status requirements.

***Example 2:***

Suppose the isolator 4 was to be prevented from operating when isolator 3 is closed or when circuit breaker B1 is closed. This condition is expressed as follows.

$$4: 3(\text{ON}) \mid \text{B1}(\text{ON})$$

***Example 3:***

Suppose that isolator 4 was to be prohibited from operating when isolator 3 or circuit breaker B1 was closed but not when 1, 2 and B5 were closed. This condition is expressed as follows.

$$4: (3(\text{ON}) \mid \text{B1}(\text{ON})) \& (! (1(\text{ON}) \& 2(\text{ON}) \& \text{B5}(\text{ON})))$$

***Example 4:***

Suppose that the circuit breaker B1 was to be prohibited from closing if Source1 was either Down, or Up and Unhealthy. This condition is expressed as follows.

$$\text{B1: } \text{B1}(\text{OFF}) \& ( \text{Source1}(\text{OFF}) \mid \text{Source1}(\text{ON UNHEALTHY}) )$$

## **5.7. Summary**

The major components of a technique for developing switching schemes have been presented in this chapter. An approach to store and access the status of equipment has been outlined. A method for storing the topology information of the network and utilizing the information to update the status of components has also been described. A load-flow calculation technique, a method for determining alternate power flow paths, and the rules for determining switching sequences have been described. Finally, generalized techniques for interlocking and sequence switching have been presented.

## **6. IMPLEMENTATION**

The concepts presented in Chapters 4 and 5 were implemented in an application development tool called Substation Switching Scheme Development Package (SSP). The tool demonstrates the feasibility of the concepts described in the previous chapters for their practical implementation.

The programming technique and the programming language chosen for the development directly affect the implementation of the technique described in previous chapters. The approach used in the development of SSP is presented in this chapter. An object-oriented approach was used to model the components of substations and switching schemes, and to develop the user-interface for the SSP. C++ language was used for writing the code.

### **6.1. Tools and Techniques**

Traditionally, algorithmic approaches are used to develop software applications. In an algorithmic approach, many software modules assist a main program that has a top-down flow. The development starts by representing the process in flow charts. These techniques are suitable for solving problems whose flow can be easily envisioned. The SSP was developed using object-oriented development technique [38-41], which facilitates the development of the software applications that are easier to maintain compared to algorithmic applications.

Any object oriented programming language, such as C++, SmallTalk, Object Pascal, or

CLOS, could have been used to write the software application. The Microsoft Visual C++ was used for reasons presented in Appendix B.

## 6.2. Substation Modelling

A substation switching system was decomposed into subsystems that in turn have their own subsystems. The decomposition was continued until some lowest component-levels were reached. For example, Status Management Units, Interlock Management Units, Command Handler Units, switches and nodes were treated as elementary objects. In this section, a convention for representing objects and classes of objects, decomposition of the system and the relationships between classes of components are discussed.

### 6.2.1. Convention

Software templates, which are generally called classes, were prepared to facilitate the modelling of components of substations. A software model of a component based on its template is referred to as an object. Each object was considered to be composed of two parts, data and functions as is shown in Table 6.1. The data represent the parameters of the object and the functions represent the behaviour of the object. The functions interpret and manipulate the data, accept commands from other objects, provide appropriate response and send commands to other objects.

Table 6.1. Structure of a class	
Class Name : ClassName	
Member Variables	Member Functions
Var1, Var2, ....., VarN	Function1, Function2, ....., FunctionN

Using the same word to denote an object and a class-of-objects can sometimes be confusing. For example, it may be confusing to use “switch” to refer to a specific switch and to a switch-class. To discriminate a class from an object, the class name is started with an upper-case character. The names of objects of that class are started with lower-case “o.” For example, if Sw denotes a switch class, oSw denotes a switch.



The names oStn and oSys are reserved for a substation and a power system respectively.

A variable or a function of a particular object is expressed by the name of the object followed by a period and the name of the variable or function. For example, the **closingTime** variable and **open** function of a switch object oSw are expressed as follows.

**oSw.closingTime**

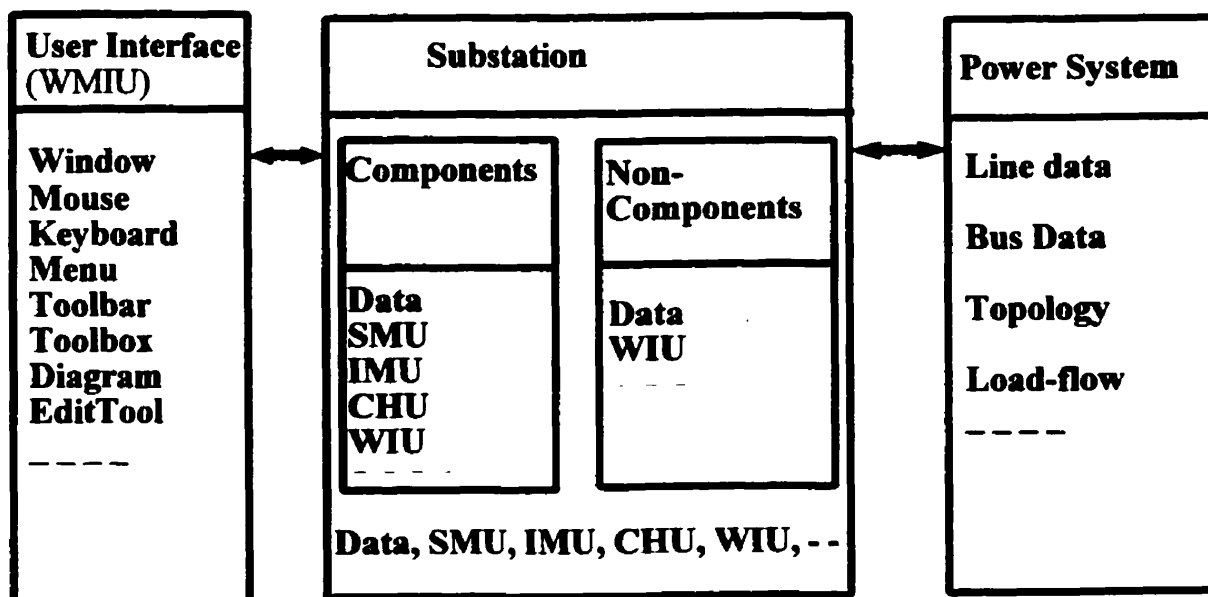
**oSw.open()**

### **6.2.2. Composition of a switching system**

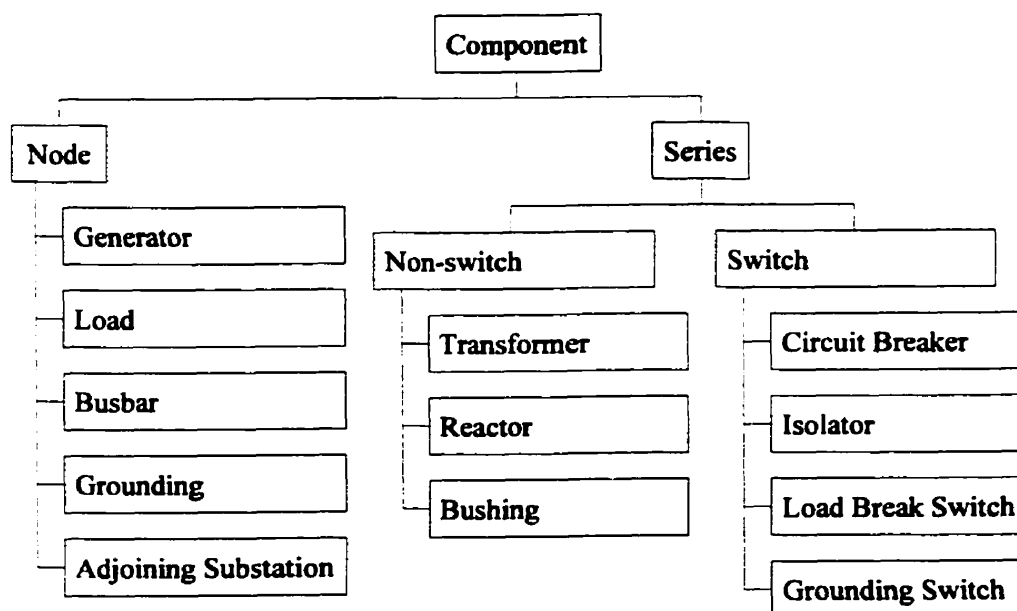
A switching system was considered to be composed of a User-Interface, a Substation-object and a Power-System-object [42-44]. The User-Interface contains all the objects required to draw and display the information in a graphical environment. The Substation-object contains all the objects such as switches, transformers, and busbars that usually reside in a substation. The Power-System-object contains power system data, system topology and load-flow modules. The User-Interface, Substation-object and Power-System-object interact with each other to simulate a switching scheme. A Substation-object interacts with the Power-System-object by sending its status parameters and receiving the result of the load flow analysis.

A substation was treated as a collection of component and non-component type objects. The components, such as switches, transformers, busbars and generators, participate in switching and control activity. The non-components, such as selected text displays, password-protection units, data-files and data access units, play supporting roles. Each component-object also has a Status Management Unit (SMU), an Interlock Management Unit (IMU), a Command Handler Unit (CHU) and a Windows Interface Unit (WIU). This relationship is shown in Figure 6.1.

The components can be of Node or Series type. A Node class includes components like loads, generators, input-output lines, busbars. A series class includes components like switches and transformers. This sub-classification of component class is shown in Figure 6.2. The list of important classes in the SSP is provided in Appendix C.



**Figure 6.1.** Composition of a switching system.

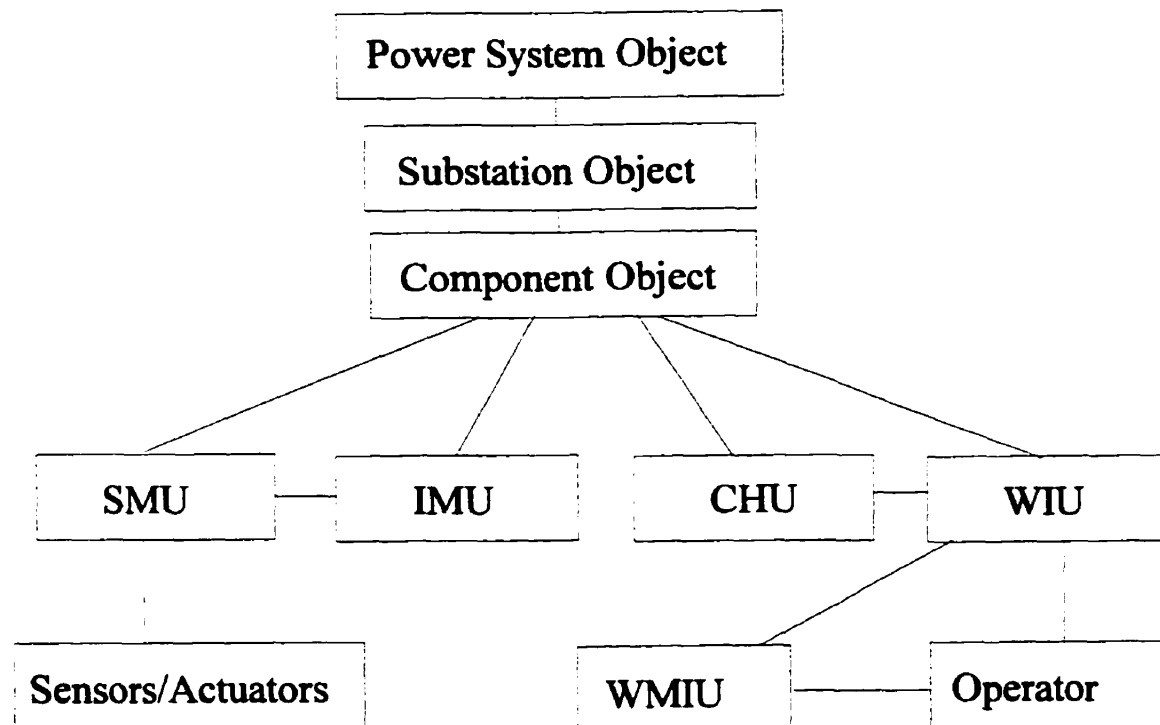


**Figure 6.2.** Decomposition of components into specialized classes.

### 6.2.3. Component relationships

Objects exchange messages for reasons such as determining the effects of a opening or closing of a switch, exchanging status information, and permitting or refusing to execute a command. Figure 6.3 shows the flow of messages between various elementary modules and a Component-object. The Component-objects pass messages with substation and with their SMU, IMU, CHU and WIU directly. The communication is indirect with the other module. A component can acquire the power system information through the substation object.

A SMU looks after the status of the component and communicates with the sensors, actuators and measurement units. An IMU evaluates the interlocks of the component. A CHU interprets and forwards the commands to the Component-objects. A WIU communicates with the user and the Windows Main Interface Unit (WMIU) that contains drawing and editing modules including toolbox, toolbar and menu.



**Figure 6.3.** Relationship of a component with various modules.

### 6.3. Data Compilation

Two types of data are handled by the SSP. The first type is the data that does not change until a physical change occurs in the substation. The second type of data changes during the normal operation of the substation.

The static-data are supplied to a switching scheme during its development. They are acquired from sources such as off-line measurements, manufacturer's data sheets, and spatial relationships of the component with other components. The dynamic-data include status parameters of components, bus-voltages, line currents, and topology information. The data are updated at regular intervals and after performing a switching operation.

A data compiler is included in the SSP for compiling the static-data and initializing the dynamic data. It also establishes communication links among objects so that they do not have to rely on the data compiler after the data is initialized.

The functions performed by a data compiler are as follows.

1. Check if all the components have the minimum required data.
2. Find the number of links, *nlink*, in the substation network.
3. Create an integer array, called *countLookup*, of length *nlink* and initialize each item to zero.
4. Count the number of items connected to each link and place the counts in the *countLookup* array.
5. If a link represents a terminal node such as a generator or a load, consider the link as a node.
6. If the count in a *countLookup* element is more than 2, consider the link associated with that element as a node.
7. Delete the *countLookup* array.
8. Create a node type object (oNd) corresponding to each node and add to the object list in oStn.

9. Assign a switch-number to each switch and a node-number to each node.
10. Sort the object list by type of objects.
11. Mark the object address of the first entry of each type of object. This would make it possible to jump from one type of object to another without traversing the list sequentially.
12. Identify the branches in the network.
13. Determine and assign the type of each branch and add the branches in the object list of oStn.
14. Establish interlock relationships among the components in each branch.
15. For each node except for the grounding node, identify adjoining branches and store in its list of adjoining branches called adjBr.
16. For each node except for the grounding node, identify adjoining nodes and store in its list of adjoining nodes called adjNd.
17. Find the adjoining nodes of transformers and store in the data of transformers.
18. Using the nominal kV of a specified node, traverse along the nodes of the substation and assign the nominal kV to all nodes.
19. Update topology.
20. Find and fill the adjoining input-output nodes of each busbar.
21. Find alternate busbars of each busbar.
22. Identify the substation configuration type and store in oStn.
23. If the priority of a busbar is not specified by the user, set it to 1 provided at least one circuit breaker exists in each path radiating from the busbar to the input-output nodes. Otherwise, set the priority to 2.
24. Interpret text based interlock expressions entered by the user, convert them to binary code and store them in the interlock management unit of the components.

25. Find default interlocks, which can be overridden by the user if necessary. The default interlocks are normally practiced rules in a typical configuration.
26. Create a link between the Load Flow Module, oLf, and the substation module, oStn.

This is a list of major functions performed by the data compiler. Many minor functions that are performed by the data-compiler are not included in the list.

## **6.4. Graphical User Interface**

A graphical user interface (GUI) for the SSP was developed independently, before developing other modules. The approach used in the development was to identify and define a set of interfaces through which other modules of SSP can interact with the components of GUI. The rest of the application program need not change when the internal implementation of the GUI changes as long as the established public interfaces do not change. This decoupling permits the visual appearance of the Windows interface to be enhanced in the future without interfering with the integrity of the SSP.

The GUI interface is prepared around WMIU, which includes a menu, a toolbar and a toolbox, as shown in Figure 6.4. The menu provides necessary command-interfaces to use all available features of SSP. A toolbox includes symbols of various substation components, which can be selected, dragged and dropped to draw single line diagrams of substations. The toolbar provides shortcut to some frequently used commands.

All the commands issuable to the SSP are organized in the following ten categories:

### **File, Edit, Operate, Control, Draw, Build, Plot, Window, and Help**

Explanation of these menu categories are given in the SSP User's Manual [45]. The tool bar provides the shortcut to the following commands that are issuable from the menu:

**File-New, File-Open, File-Save, Edit-Cut, Edit-Copy, Edit-Paste, Edit-Delete, Edit-Undo, Draw-Edit Substation Diagram, Plot-Voltages and currents, Build-Compile Component, Compile, Build-Compile Substation, File-Print, and Help-Index**

The toolbox presents a set of symbol-buttons, which can be pressed to activate a WIU of a new instance of the selected component class. The WIU allows the user to draw the component on the screen, and provide the input data of the component. When a component symbol is added (or removed) on the computer screen, an object is created for (or deleted from) the switching scheme.

A component is selected by clicking the left mouse button over it. Dialog boxes to edit data, to view data or to issue commands are activated by selecting the component and selecting appropriate menu items. More discussion on WIUs is provided in Appendix D.

#### **6.4.1. Role of WIU**

A WIU handles five types of windows based activities. They are as follows:

1.     **Symbolic Display:** Each physical component of a substation is depicted on the screen as a part of the substation diagram.
2.     **Data Display:** The component data are displayed on the screen by using the data display windows. These windows are dialog boxes specifically designed for each class of object that display the data in the appropriate format.
3.     **Data Edit:** Each WIU has its associated data-dialog box which displays the component data and allows the user to edit the data. The dialog boxes are used mainly for editing the properties of the components.
4.     **Error Report:** When an error is encountered while processing the component data, a message is displayed in an error-dialog box.

5. **Command Interface:** Each WIU has its associated command-dialog box that displays the list of all relevant operating commands. The command selected by the user is identified and sent to the **Command Handler Unit**.
6. **Status Update:** Each WIU has an associated status-dialog box that allows the user to change selected status parameters.

All component objects have these features. However, only a subset of these features are available for a non-component object.

#### **6.4.2. Tool box and substation diagrams**

A toolbox, as shown in Figure 6.4, is activated by selecting **Edit**, and then **Launch Editor**, from the menu of the WMIU. The set of symbols shown in the toolbox are used to draw substation diagrams. The **Cancel** and **Ok** buttons shown in the toolbox are used to exit from the tool box and, consequently, from the editing mode of the software.

When a symbol is selected from the toolbox and the mouse is moved to the drawing area, the symbol is drawn at the location where the mouse is clicked. The WIU allows the objects drawn on the screen to be moved, rotated, shifted in selected step-sizes, copied, pasted and deleted. The font, size and colour of the text can be changed. The procedures that a user must follow to draw a substation diagram are given in the **SSP User's Manual [45]**.

Public interfaces, such as the following, are provided in WIU's of substation components through which rest of the SSP modules interact with the graphical elements:

**Operate(), ChangeProperties(), Compile(), Draw(), Select(), Rotate(),  
Shift(int dx, int dy), Copy(), Paste()**

The switching components also include the following two important public interfaces:

**Open()  
Close()**



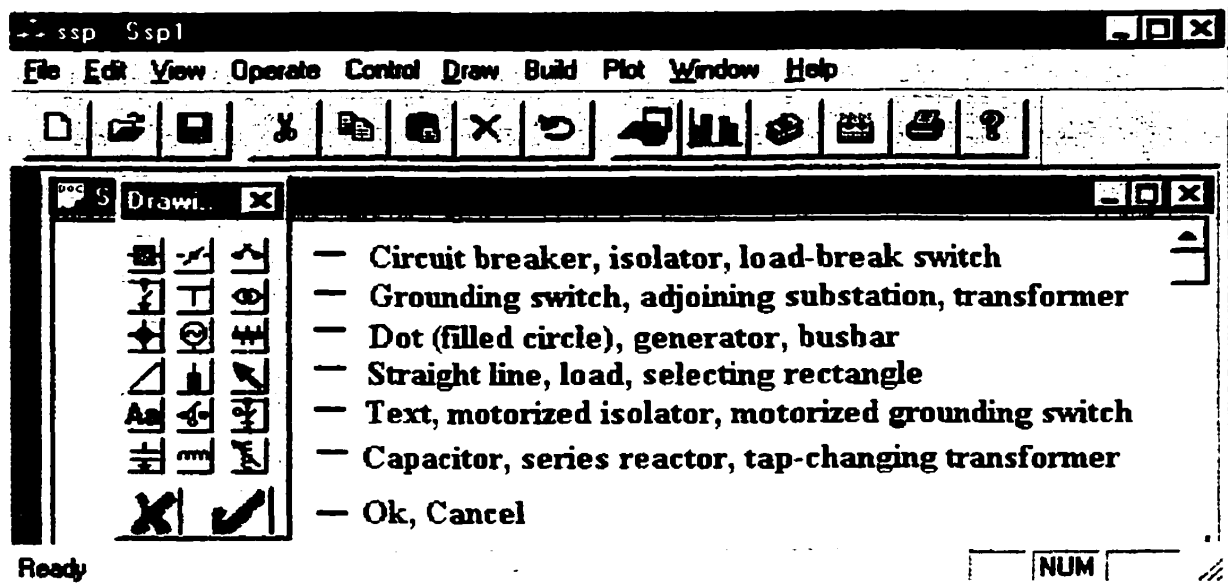
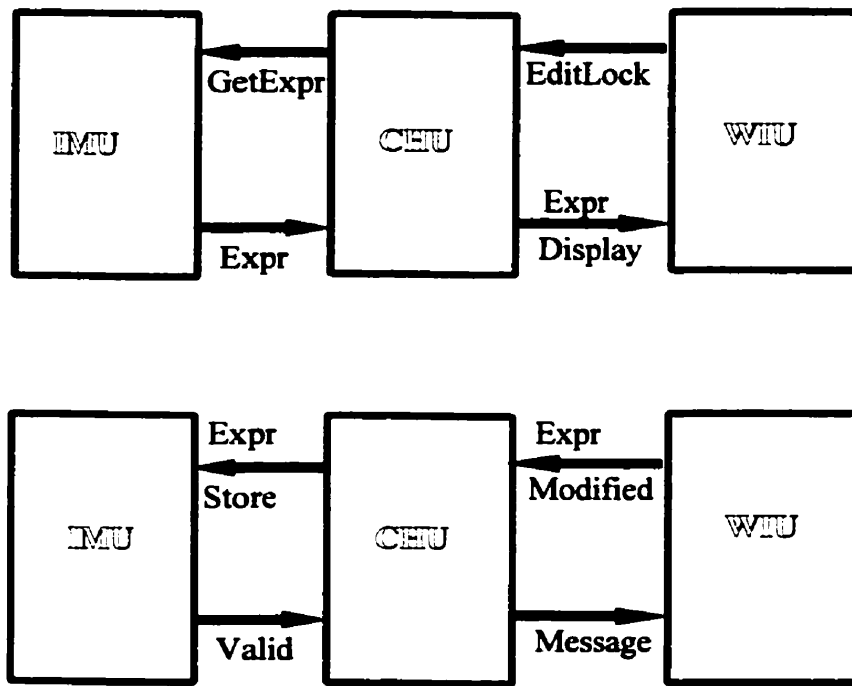


Figure 6.4: The toolbox.

## 6.5. Editing Interlocks

This section explains the implementation of the facility for editing interlock-expressions for component-objects. The message-flow that happens during the process of editing the interlock expressions is shown in Figure 6.5.

A WIU asks for the interlock expression of the component from the CHU, which in turn acquires the interlock expression from the IMU and provides it to the WIU. The WIU displays the interlock expression and allows the user to make modifications. However, a WIU cannot determine whether the user entry is valid or not. The WIU sends the modified expression to the CHU that passes it on to the IMU. The IMU stores the expression if it is valid, but sends a warning message to the WIU, via the CHU, if it is invalid.



**Figure 6.5:** Sequence of events in interlock modification.

## 6.6. Implementation of Status Management Units

The *uniform status storage and processing protocol*, defined in Chapter 5, required that each component-object must have a status management unit. The status variable contained in the SMU is an integer, which provides bit fields for storing the status parameters. Bitwise operations of integer variables is performed to check the parameters. The bitwise approach was used to expedite the comparison of status parameters.

Some functions that were defined for checking all or some parameters stored by SMU using a single command are as follows:

**Status** GetStatus()

**String** GetStatusString()

**String GetKeyString()**

**int CheckAll( Status teststs ) {return sts == teststs;}**

**int CheckSelective(Status teststs, Status key) {return (sts & key) ^ teststs;}**

In these functions, **sts** represents the status variable stored in SMU, **teststs** is the status to be compared with **sts**, and **key** is an input parameter that indicates the bit-fields to be tested.

The **CheckAll** function compares the status parameters with the input parameters provided by the user. The result is 1 if all the parameters in **sts** and the input (**teststs**) are identical.

The **CheckSelective** function checks selected fields only. It takes two parameters, called **teststs** and **key**, as inputs. The value of the function is returned. If this value is zero, it is interpreted as **TRUE**, otherwise **FALSE**.

Consider an example in which **sts**, **teststs** and **key** are as follows.

**sts = 1011100000001111 1011100000001111,**

**teststs = 0000000000101110 0000000000101110, and**

**key = 0000000000101111 0000000000101111.**

The **key** tells that only the first four bits and the sixth bit should be compared. However, bit 1 and bit 6 of the **sts** and **teststs** are not identical. Therefore the **CheckSelective** function returns a non-zero value. The procedure for executing this function is as follows.

1011100000001111	1011100000001111	(sts)
0000000000101111	0000000000101111	(key)
<hr/>		Bitwise-AND
0000000000001111	0000000000001111	
0000000000101110	0000000000101110	(teststs)
<hr/>		Bitwise-XOR
0000000000100001	0000000000100001	(> 0)

Conclusion: FALSE

A number of other functions for checking status one parameter at a time are as follows:

**On(), Locked(), MakeI(), BreakI(), Free(), ManLocked(), RidDefault(),**

**Bypass(), NormI(), ExtLive(), RemoteIsol(), Loaded(), Transit(), NormV(), OffLimitV(), Enabled(), Healthy(), Repair(), Isolated(), Grounded(), SCPath(), NormGenrn()**

This set of functions returns the concerned bit directly. For example, On() would return 1 if the device were “on”, otherwise it would return 0.

The on-line update of the status is done in conjunction with sensors/actuators and user interface modules. These peripheral units send the changes encountered in physical devices or user options, by using dedicated interface functions. The SMU analyses the input messages and sets the status fields accordingly.

## **6.7. Implementation of Interlock Management Units**

The main components of an *Interlock Management Unit* are the interlock expressions (LockStrings), an expression-interpreter, an interlock-check module and the required number of *Lock Units*. The implementation of these modules are presented in this section.

### **6.7.1. Implementation of Lock Units**

*Lock Units*, described in Section 5.8, form the building blocks for writing interlock expressions. A *Lock Unit* consists of three variables named **componentRef**, **teststs**, and **key**.

Consider the following text-based expression translated into a lock unit.

**Breaker1(ON ENABLED UNLOCKED)**

The address of **Breaker1**, given in the list of objects contained in oStn, is stored in **componentRef**. The **key** is set to indicate that ON, ENABLED and LOCKED parameters are to be tested. The variable **teststs** is set to indicate that ON, ENABLED and LOCKED parameters must be 1, 1 and 0 respectively.

### 6.7.2. Compilation of text based expressions

The text based expressions, which are in infix notation, are translated into postfix notation. Two examples of such expressions are as follows.

**a | b** (infix notation)

**a b |** (postfix notation).

In these examples, a and b are operands, which would be LUs in this application, and the symbol is the OR operator.

A fully parenthesized infix expression can be translated by starting from the innermost parenthesized sub-expression and proceeding outwards. For example, to convert the expression

**(A | (B|C) & D) )**

to the postfix notation, the innermost sub-expression (B|C) is first converted to B|C|. The sub-expression ((B|C)&D) is then converted to B|C|D&|. Finally, the expression (A | B|C|D&|) is converted to A|B|C|D&|. It is interesting to note that the conversion does not change the sequence of the variable names. The algorithm developed for this conversion is given in Appendix E.

A postfix expression is evaluated by scanning it from left to right. The operators observed in this scan are implemented as encountered. For example, to evaluate A|B|C|D&|, the expression is scanned as shown in Table 6.2.

<b>Table 6.2. Evaluation of a postfix expression</b>			
<b>Expression</b>	<b>Observed Operator</b>	<b>Operands</b>	<b>Result</b>
ABC D&		B, C	T <sub>1</sub>
AT <sub>1</sub> D&	&	T <sub>1</sub> , D	T <sub>2</sub>
AT <sub>2</sub>		A, T <sub>2</sub>	T <sub>3</sub>

To detect syntax errors, postfix expressions are viewed to be composed of a set of LUs and a set of operators. A concept of **rank**, with the following definition, is used to detect errors.

1. The rank,  $r$ , of a LU is 1.
2. The rank of a binary operator is -1 and that of an unary operator is 0.
3. The rank of a sequence of LUs and operators is the arithmetic sum of the ranks of LUs and operators;

For example  $r(abcd\&!)=r(a)+r(b)+r(l)+r(c)+r(d)+r(\&)+r(!)+r(!)$   
 $=1+1-1+1+1-1-1-0=1$ .

This postfix expression is valid because its rank is 1.

4. A postfix expression is valid only if its rank is equal to 1.

### **6.7.3. Evaluation of interlocks**

The interlock expressions (LS, DLS or OLS) stored in an IMU of a component are based on postfix notation as described in Section 6.7.2. The method of evaluating an interlock expression is as follows.

1. Find the leftmost operator in the expression.
2. Obtain the two LUs which immediately precede the operator just found if it is a binary operator. Obtain one operand which immediately precedes the operator just found if it is a unary operator.
3. Take the first LU, and evaluate. The evaluation produces a non-zero value to represent TRUE and 0 to represent FALSE. Keep the result in a temporary variable, temp1. If applicable, evaluate the second LU and keep the result in another variable, temp2.
4. Perform the indicated operation on the temporarily stored values, temp1 (and temp2 if applicable) and obtain the result.
5. Replace the operator and the LUs with the result.

6. Go to Step 1 if there is an outstanding operator in the expression.

This procedure is used to evaluate all the expressions that might exist in an IMU.

#### **6.7.4. Determination of locked state**

The following algorithm was developed and used for evaluating the LOCKED/UNLOCKED status of a component.

1. Check if MANLOCK is TRUE. If it is TRUE, ignore the rest of the steps and set LOCKED parameter of SMU to TRUE.
2. Assign FALSE to missing LS and DLS expressions in the IMU. Assign TRUE to a missing OLS.
3. Set LOCKED parameter of the SMU to TRUE if  $(OLS \& (DLS \mid LS))$  is TRUE. Otherwise, set LOCKED parameter of the SMU to FALSE. This relationship is shown in Figure 6.6.

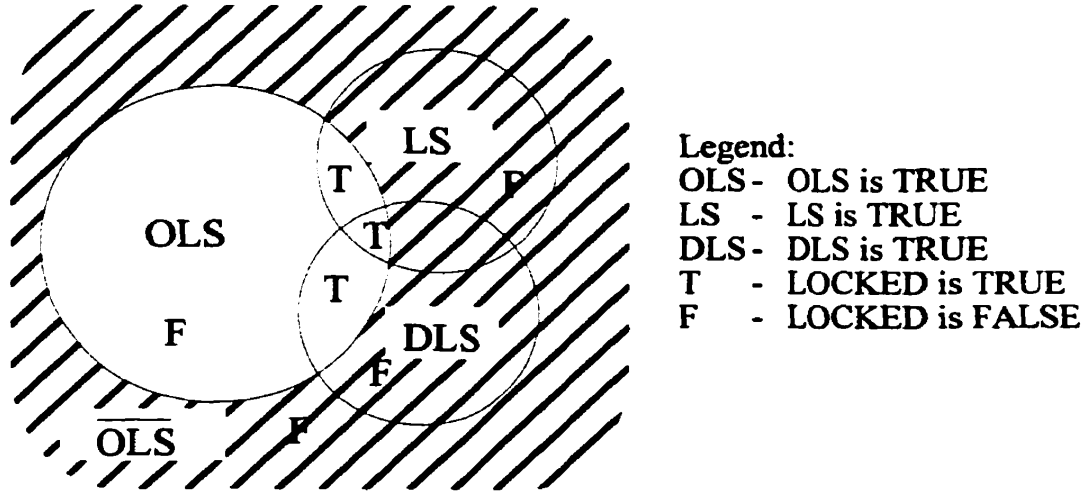
A component that is not locked may still be inoperable for several reasons, such as being in an unhealthy state.

### **6.8. Implementation of Command Handler Unit**

The Command Handler Unit (CHU) of a component-object handles five types of commands.

- i. Symbol Manipulation Commands, for example Move, Rotate, Delete, Create.
- ii. Data View Commands, for example Display-data.
- iii. Data Edit Commands, for example Edit-data.
- iv. Status Update Commands, for example Enable, Disable, Manlock.
- v. Operate Commands, for example Open, Close, Isolate.

The CHU of a component receives these commands either from another component or from the user through the GUI. The implementation of these commands requires several other supporting modules such as topology processor, load-flow analyser, IMU and SMU.



**Figure 6.6.** Logical relationship of DLS, LS and OLS with LOCKED status.

### 6.8.1. Switching operations

The handling of important switching commands is described in this section.

#### 6.8.1.1. Close or open a switch

The following procedure was developed for operating a switch, oSw, that belongs to a branch, oBr.

1. If oSw.Disabled or oBr.Disabled is TRUE, print a message and return.
2. If oSw.Locked or oBr.Locked is TRUE, print a message and return.
3. If oBr.NoOnOff is TRUE, close/open oSw, update data and return.
4. Implement the function oStn.TempTopology.
5. Implement the function oStn.TempMarkChargedIONodes.
6. If oBr.SCPATH is TRUE, close/open oSw, update data and return.
7. If the command is for closing perform the following operations.
  - i. If oStn.GroundsLive is TRUE, print a message and return.
  - ii. If oSw.IsIsolator and oStn.MakesCurrent are TRUE, print a message



and return.

If the command is for opening perform the following operations.

- i. If oSw.IsIsolator and oStn.BreaksCurrent are TRUE, print a message and return.
8. Implement the function oSys.LoadFlow.
9. If oSys.OffLimitVI is TRUE, print a message and return.
10. Close (or Open) oSw, update data, print a message and return.

#### ***6.8.1.2. Connect a node to another node***

This command is issued to a node, which is identified as the start-node (oNd1). The start-node is to be connected to a destination-node (oNd2). The following procedure was developed for this purpose.

1. If start and destination nodes are the same, print a message and return.
2. If oNd2.Disabled or oNd2.Locked is TRUE, print a message and return.
3. If oNd1 and oNd2 are already connected, print a message and return.
4. Implement the function oNd1.SearchPaths.
5. Implement the function oStn.CloseBestPath.

#### ***6.8.1.3. Disconnect a node from another node***

This command is issued to a start node, oNd1, for disconnecting it from the destination node oNd2. The following procedure was developed for disconnecting a node from another node.

1. If the start and destination nodes are the same, print a message and return.
2. If oNd2.Disabled or oNd2.Locked is TRUE, print a message and return.
3. If oNd1 and oNd2 are already disconnected, print a message and return.
4. Implement the function oNd1.SearchPaths.
5. Implement the function oStn.OpenAllPaths.

When disconnecting two nodes, all the paths leading to the destination node are opened.

#### ***6.8.1.4. Isolate node***

The following algorithm was developed for isolating a node, oNd1, from the system.

1. Make a nodeList containing input-output nodes and busbars connected to oNd1.
2. Find paths radiating out from oNd1 until each path encounters an isolator.
3. Connect nodes in the nodeList to one another without using oNd1 and isolators identified in Step 2.
4. If all nodes of nodeList are connected to one another, open all circuit breakers and then isolators that are in the paths found in step 2.

#### ***6.8.1.5. Isolate a series component***

The following algorithm was developed for isolating a series component oSe contained in a branch oBr.

1. If there is no isolator between the component and one end, End1, of oBr, implement the following function.

End1.Isolate
2. If there is no isolator between the component and the other end, End2, of oBr, implement the following function.

End2.Isolate
3. If End1.Isolated or End2.Isolated is True, open the circuit breaker and then isolators contained in oBr and return.
4. If oBr.SCPATH is TRUE, open the circuit breaker and then isolators contained in oBr and return.
5. Make a nodeList of input-output nodes and buses connected to End1.
6. Connect nodes contained in the nodeList to one another without using oBr.
7. If all nodes contained in the nodeList are connected to one another, open the

circuit breaker and then isolators contained in oBr and return. Otherwise, print a message and return.

#### **6.8.1.6. Bus transfer**

The following algorithm was developed for transferring circuits from one busbar, oBus1, to another busbar, oBus2.

1. Make a busList containing busbars connected to oBus1.
2. Make a nodeList containing input-output nodes that are connected to oBus1.
3. Connect busbars contained in busList to oBus2 without using oBus1.
4. Connect nodes contained in nodeList to oBus2 without using oBus1.
5. If all nodes of nodeList are connected to oBus2, open all branches that are connected to oBus1.

#### **6.8.2. Supporting modules**

Supporting modules provide vital information to CHUs. The most frequently used supporting modules include a topology determination module, a module to predict the status of the input-output nodes, and a module to trace paths from one node to another node.

##### **6.8.2.1. Topology tracer**

The algorithm described in this section was developed to trace the topology of a substation. Before the topology is traced, the circuit must have been successfully compiled. The software would have identified the branches and would have prepared the lists of adjoining nodes and adjoining branches. The developed topology tracer algorithm consists of the following steps.

1. Set a busbar as the start node, *startNode*.
2. Declare two pointers called *current* and *neighbour* to identify nodes.
3. For every node, oNdx, of the substation, initialize the following variables.

- oNdx.reached = FALSE (indicating oNdx is not visited)
  - oNdx.oldTopo = oNdx.newTopo
  - oNdx.newTopo = oNdx.nodeNumber.
4. Prepare an empty *queue* of nodes.
  5. Add the *startNode* to the queue and set startNode.reached as TRUE.
  6. Remove the first element from the *queue* and assign it to *current* node.
  7. For each adjoining node of *current* node perform the following operations.
    - a. Assign the selected adjoining node to *neighbor* node.
    - b. If the *branch* connecting the *current* node and the *neighbor* node is ON and the *neighbor* was not previously visited, do the following.
      - i) If  $\text{current.newTopo} > \text{neighbor.newTopo}$ , do the following.
 

$$\text{current.newTopo} = \text{neighbor.newTopo}$$

$$\text{current.reached} = \text{FALSE}$$

Otherwise, set  $\text{neighbor.newTopo} = \text{current.newTopo}$ .
      - ii) Add *neighbour* to the end of the *queue* and set neighbor.reached to TRUE.

Otherwise, if the branch connecting the current node and the neighbour node is ON, the newTopo of current node and neighbour node are not equal, and the neighbour node was previously visited, perform the following operations.

    - i) Set the newTopo of the nodes as follows.
 

$$\text{neighbor.newTopo} = \text{Min}(\text{current.newTopo}, \text{neighbor.newTopo}).$$

$$\text{current.newTopo} = \text{neighbor.newTopo}.$$
    - ii) Add *neighbour* to the end of the *queue*.
  8. If the *queue* is not empty, proceed to Step 6.

#### **6.8.2.2. Energized node detector**

The following algorithm was developed to detect whether a node is energized or not. Before the detection process is started, the input-output nodes connected to live sources must be known.

1. Declare an empty *queue* of nodes.
2. For each input-output node, *oNdx*, perform the following initialization.
  - If the node is, or is connected to, a live source, do the following.
    - Set *oNdx.On* to TRUE and add the node to the *queue*.
  - Otherwise, set *oNdx.On* to FALSE.
3. For each node that is not an input-output node, *oNdy*, set *oNdy.On* to FALSE.
4. Identify all groups of nodes that are connected to each other and are in the *queue*. Retain one node from each group and remove other nodes from the *queue*.
5. From the *queue*, remove the node at the top, *startNode*.
6. Make a *queue2* and add the *startNode* to the *queue2*.
7. From *queue2*, remove the node at the top, *currentNode*.
8. For the *currentNode*, take the first adjoining node, *neighborNode*, that was not previously visited.
9. If the current node and the neighbour are connected, do the following.
  - i) Set *neighborNode.On* to TRUE.
  - ii) Add *neighborNode* to the bottom of the *queue2*.
10. If *currentNode* has an adjoining node that was not previously visited, proceed to Step 8.
11. If the *queue2* is not empty proceed to Step 7.
12. If the *queue* is not empty proceed to Step 5.

### **6.8.2.3. Detector of the making or breaking of currents**

The following procedure was developed to detect whether a switching operation makes or breaks currents.

1. Implement the function `oStn.TempTopology`.
2. If the ground node is ON, implement the function `oStn.GroundsLive(True)`.
3. Make a *nodeList* containing input-output nodes.
4. Take the topmost unvisited node `oNode1` from *nodeList*.
5. From `oNode1` to the end of *nodeList*, sequentially take one node at a time and call it `oNode2` and do the following.

If `oNode1` or `oNode2` is energized then do the following.

If `(node1.newTopo = node2.newTopo)` and

`(node1.oldTopo <> node2.oldTopo)`, implement the following function.

`oStn.MakesCurrent(TRUE)`

Otherwise, if `(node1.newTopo <> node2.newTopo)` and

`(node1.oldTopo = node2.oldTopo)`, implement the following function.

`oStn.BreaksCurrent(TRUE)`

6. Set `oNode1.visited` to `TRUE`.
7. If `oNode1` is the second last element of *nodeList*, stop the process. Otherwise, proceed to Step 4.

#### **6.8.2.4. Path tracer**

The following algorithm was developed for tracing paths from a node, *startNode*, to destination nodes, *destNode*.

1. Declare two node type object pointers called *neighbourNode* and *currentNode*.
2. Declare an empty *queue* of nodes.
3. Empty the *result*, which is contained in oStn.
4. Mark all nodes in the network as not visited.
5. Set the *startNode.visited* to TRUE and place it in the *queue*.
6. Remove a node from the top of the *queue* and assign it to *currentNode*.
7. If the *currentNode* is a destination node, *destNode*, append it to *result* and proceed to Step 10.
8. Get the first adjoining node that was not previously visited, *neighbourNode*, and do the following operations.
  - i) Set *neighbourNode.visited* to TRUE.
  - ii) Set the *currentNode.parent* to *neighbourNode*.
  - iii) Add the *neighbourNode* to the end of the *queue*.
9. If the *currentNode* has an adjoining node that was not previously visited, proceed to Step 8.
10. If the *queue* is not empty, proceed to Step 6.

### **6.9. Integrating with System Data**

Switching actions taken in one substation change the voltages and current flows in the entire power system. An improper operation in one substation can have adverse effects on a large part of the network [46, 47]. The prediction of the effects of a switching operation

is possible only if the switching scheme evaluates the system performance. The ability to analyse system data was, therefore, included in the SSP. This facility checks that the security of the system is not violated while switching operations are performed.

A switching plan is first evaluated at the substation level. If the plan is feasible, a load flow is conducted to confirm that voltage and current flow limits are not violated..

#### **6.9.1. Load flow data formats**

SSP recognizes two formats for receiving load-flow data. Both formats are based on ASCII text. One of the formats is the “IEEE Common Data Format for the Exchange of Solved Load Flow Data,” which is also called CDF format [48]. This format provides detailed information of the system but requires large data files. Many items included in this format are beyond the scope of the SSP. The advantage, however, is that this format is known to, or used by, many utilities for exchanging load-flow data. The second format was developed for use by the SSP; this format is named “Reduced Data Format,” or RDF. It contains a subset of the data in the CDF format. The specifications of these formats are given in Appendix F. Sample data and diagram of a 9 bus test power system expressed in RDF are shown in Appendix G.

#### **6.9.2. Linking the substation data with the load-flow data**

The load flow data and the substation data are of different nature. A transmission line connecting one substation to another is treated as an input-output node by the substation-objects, whereas the line is treated as a branch by a load flow object. The network that is beyond the boundary of the substation is not retained by a substation-object. The effect of the external network is considered by connecting equivalent load or generation at the input-output node of the substation. The details of the connections of the circuit components in the substation are retained. The load flow data, on the other hand, lumps all the components in a substation and treats the substation as a node. The load-flow data treats the substation as a busbar connected to other busbars of the system via transmission lines, which are considered as branches of the power system network.



The SSP links the substation objects and the system objects and facilitates the exchange of information between them. Some of the information exchange takes place through files and some through public functions of the software objects.

The following information is sent to the **load-flow analyser** object by the **substation** object directly through the interface functions.

1. The list of lines that are disconnected from the substation.
2. The list of lines that are connected to the busbars of the substation.
3. The name of the files where the substation-object wants the Load-flow analyser to store the load flow results.
4. The bus number by which the substation is identified in the load flow data.
5. Total active power and reactive power drawn from, or injected into, the substation.

The load flow analyser updates its data to reflect the connection of the lines and busbars. It then performs the load flow analysis and returns the results in the specified files. The SSP reads the data and uses it to check the violations of voltage and current levels.

### **6.9.3. The role of the user in the integration**

The integration of the substation switching scheme with the load flow analyser is largely automated. However, the user must provide some information for proper integration. The user plays the following roles.

1. Provide the load flow data of the system in either CDF or RDF format.
2. Examine the load flow data, determine the bus identification numbers of the adjoining substations and identify the branches that connect the substation to the adjoining substations.
3. For each adjoining substation object, perform the following operations:
  - a. Provide the identification number used by the load flow data for the

- line to the selected adjoining substation.
  - b. Provide the bus identification number used by the load flow data for the adjoining substation.
4. Open appropriate interface for entering system properties.
  5. Provide the assigned system bus number.
  6. Specify the files where the input data for power flow calculations are stored.
  7. Specify the files where the power flow output data are to be stored.
  8. Recompile the switching scheme. Choose to integrate the load flow with the switching scheme.

Once the switching scheme is compiled, the SSP automatically handles the data.

## **6.10. Graph Plotting**

The process of analysing data in a substation environment is tedious, time consuming and error prone, particularly when the data is not presented in an appropriate graphical format. The complexity of analysis grows with the size of the data. It is, therefore, valuable to provide an interactive tool that analyses and presents data in a user-friendly graphical environment. Alternative switching plans can be quickly analysed and their effectiveness can be promptly evaluated with the help of such a tool.

The graph plotting module is one of the user-productivity enhancement feature provided in the SSP. The module can plot graphs of many types of substations and power system variables so that they can be visually inspected and compared. This section describes the procedure of plotting the graphs.

### **6.10.1 Prerequisites for running the graph plotting program**

The data must be made available in files of predefined formats before launching the graph plotting program. The current version of the graph plotting program requires that the data

must meet the following requirements.

1. File names      The following file names must be used to supply data to the graph plotting program:
  - 0LocPlot.dat:**      File containing the local data of the substation
  - 0LFBusD.dat:**      File containing the magnitudes and phase angles of the voltages
  - 0LFLineD.dat:**      File containing the line data.
2. Plot module      The name of the plotting program is **plotdata.exe**. It can be installed in any directory, as long as it is specified to the SSP through system-data properties dialog box before or during the compilation process.
3. Data Directory      The data files 0LocPlot.dat, 0LFBusD.dat, and 0LFLineD.dat must exist in the same directory as the graph plotting program.

The graph plotting window will display a blank box and a random graph if the graph plotting command is issued from the SSP before the necessary data are available. The commands issued from the graph-plotting window will produce error messages on the screen.

The graph plotting program has a Text-edit box, which displays the content of the file whose data is plotted.

The graph-plotting program can be **launched** from the SSP's working environment. It can also be launched **from outside the SSP** environment by using the following procedure.

1.      Identify the directory where the main graph-plotting program is located.
2.      Make sure that the preconditions mentioned earlier in this section are met.
3.      Launch the Plotdata.exe file.

More details on using the graph-plotting program are given in the SSP User's Manual.

## **6.11. Summary**

Object oriented techniques and languages are more appropriate for developing systems, like the SSP, compared to the algorithmic approaches. The object-oriented technique can be used to decompose the problem and manage its complexity.

The switching system is a collection of circuit elements that are considered as objects. The characteristics of the components are represented by these objects. Each object consists of modules that include the Status Management Unit, the Interlock Management Unit, the Command Handler Unit and the Windows Interface Unit. The objects interact with each other and with the graphical user interface.

This implementation of the objects of the switching system and their relationship have been described in this chapter. The focus has been on the implementations of the status management unit, the interlock management unit, the user interface unit and the command handler units. The outcome of the implementation of the concepts is a software package called the Substation Switching Scheme Development Package (SSP). Its performance will be studied in the next chapter.

## **7. RESULTS**

Several tests were conducted to validate the techniques that have been discussed in the previous chapters of this thesis and have been implemented in the SSP application package. The validity of individual modules of the SSP was also tested at various stages of their development. The SSP package was used to develop switching schemes for eight configurations of substations. The suitability of the switching schemes generated by the SSP was then examined. The substations and the tests are presented in this chapter. These tests show that it is possible to implement the technique and develop switching schemes for practical configurations of substations.

### **7.1. Test Substations**

Eight substations used to test the SSP are shown in Figure 7.1 through Figure 7.8. Brief introductions of these substations are presented in this section.

#### **7.1.1. Test Substation 1**

Test substation 1, which is a double busbar substation, is shown in Figure 7.1. Main features of this substation configuration are given in Section 2.3.3. Bus1, which is the main busbar, is used during the normal operation of the substation. Bus2, which is an auxiliary busbar, is used when Bus1 has to be taken out of service. The bus coupler circuit, which consists of switches 1, 2 and circuit breaker B5, is for temporarily connecting the busbars during the switching operations.

### **7.1.2. Test Substation 2**

Test Substation 2, which is a main-and-transfer busbar arrangement, is shown in Figure 7.2. Main features of this configuration are described in Section 2.3.7. Isolators 4, 6, 8 and 10 are by-pass isolators. Isolators 1 and 2, and circuit breaker B5 form the bus-coupler circuit. As described in Section 2.3.7, it is possible to take a circuit breaker out of service without interrupting the energy flow on the circuit.

### **7.1.3. Test Substation 3**

Test Substation 3, shown in Figure 7.3, consists of two sections; each section has a different configurations. One section has a double busbar configuration that is described in Section 2.3.3. The second section has a two circuit breaker configuration that is described in Section 2.3.6. Isolators 15 and 16 are provided to connect together the two sections of the substation.

### **7.1.4. Test Substation 4**

Test Substation 4, shown in Figure 7.4, is composed of two sections of double busbar configuration. The configuration is described in Section 2.3.3. The two sections are connected by bus sectionalizing isolators 17 and 18. This simple and small substation was used to obtain many test scenarios during the initial phase of the development of the SSP.

### **7.1.5. Test Substation 5**

Test Substation 5, shown in Figure 7.5, consists of two sections of different configurations. One section has a double busbar configuration that is described in Section 2.3.3 while the other section has a main and a transfer busbar configuration that is described in Section 2.3.7. Isolators 15 and 16 connect the two sections of the substation.

### **7.1.6. Test Substation 6**

The single line diagram of Test Substation 6 is shown in Figure 7.6. This is a 765 kV substation that is connected to a 400 kV substation via 765/400/33 kV transformers. The configuration of the substation is a modification of a double busbar arrangement with a by-

pass isolator. Additional features are included in this arrangement for increasing the flexibility during the operation of the substation.

#### **7.1.7. Test Substation 7**

The single line diagram of Test Substation 7 is shown in Figure 7.7. This is a 400 kV substation that is connected to the 765 kV substation shown in Figure 7.6. This substation has a double busbar with by-pass isolator arrangement that is described in Section 2.3.5.

#### **7.1.8. Test Substation 8**

The single line diagram of Test Substation 8 is shown in Figure 7.8. This substation consists of three distinct configurations. The first configuration is a double-busbar arrangement; the second configuration is a main-and-transfer bus arrangement and the third section is a two-circuit breaker arrangement. Bus sectionalizing isolators 15, 16, 48 and 47 are provided to connect the sections together.

### **7.2. Preparation of Switching Schemes**

The switching schemes for the selected substations were prepared in four phases. The first phase consisted of preparing single-line diagrams that provided the user interface to the SSP. The second phase consisted of entering the data of the components of the substations. The SSP prepared the interlock constraints of the switching schemes in the third phase. Finally, the SSP analyzed the data and determined the switching sequences. These phases are briefly discussed in this section.

#### **7.2.1. Preparing user interface**

The details of the procedures for drawing substation diagrams that provide user interface for the substations are given in the SSP User's Manual. An overview of the procedure is given in Section 6.4. The substation diagrams are drawn in the GUI environment by using the drag-drop-and-arrange technique. The single line diagrams of the substations shown in Figures 7.1 through Figure 7.8 were prepared in this manner. These diagrams provided the user interfaces for the substations.

### **7.2.2. Entering and initializing component data**

The data of the components were entered. The edit mode of the substation diagram was first selected. The cursor was placed on an item of the diagram and right mouse button was clicked twice. This displayed a dialog box for entering the properties of the selected component. For example, the Switch Properties dialog box, which is shown in Figure 7.9, has fields that allow the user to modify the properties of Circuit Breaker C4.

### **7.2.3. Developing interlock logic**

The SSP automatically specifies interlocks for components based upon general constraints. The interlock logic for a substation, which is determined automatically by SSP, is often adequate for simple configurations, such as the configurations of Substation 1 through 5. When the configurations are complex, the people responsible for developing the interlock scheme should evaluate the interlock expressions for each component and modify them to meet the specific requirements of the substation. The constraints that are not covered by the default logic should be included. Interlock expressions for Substations 6 and 7 were checked and modified to suit the needs of the substations. The next section describes how such substation specific constraints were developed. Test Substation 6 is taken as an example to demonstrate the procedure.

## **7.3. Complex Interlocking Schemes**

The interlocks for the substation of Figure 7.6 are recorded in this section to demonstrate the procedure for developing a comprehensive interlocking scheme without using the default interlocks provided by the SSP.

The Test Substation 6 is composed of typical multi-branch segments shown in Figure 7.10. The interlock requirements of these segments are discussed in this section. The interlock requirements of circuit segments that are not of typical configuration can be determined by extrapolating the logic presented in this section.



**Circuit segment of Figure 7.10(A):** The switches of this segment are operated when the line has to be taken out of service for repair or maintenance. This necessitates that the circuit segment be discharged before the work can be started. The grounding switches are used for this purpose. It is necessary to make sure that the circuit segment is isolated from both ends before the grounding switches are closed. The interlocks defined by the following logic would ensure that the segment is isolated before the grounding switches are closed;

$$\begin{aligned} \text{G43: } & ( (\text{Bush4(FUNCTIONAL)} \& \text{7L4(FUNCTIONAL)}) \\ & \quad | \text{I44(ON)} | \text{7L4(ON)} | \text{7L4(NOREMOTEISOL)} \\ & ) \text{-----} \end{aligned} \quad (7.1)$$

and

$$\begin{aligned} \text{G44: } & ( (\text{Bush4(FUNCTIONAL)} \& \text{7L4(FUNCTIONAL)}) | \\ & \quad | \text{I44(ON)} | \text{7L4(ON)} | \text{7L4(NOREMOTEISOL)} \\ & ) \text{-----} \end{aligned} \quad (7.2)$$

The expression implies that the grounding switch must not be closed if:

- it is not for the maintenance or repair of either the bushing or the line
- the isolator I44 is closed
- the circuit is not isolated from the remote end.

The isolator I44 must not be closed before the grounding switches are opened. This interlock can be defined by the following expression:

$$\begin{aligned} \text{I44: } & ( \text{G43(ON)} | \text{G44(ON)} \\ & ) \text{-----} \end{aligned} \quad (7.3)$$

In addition to these constraints, all other constraints, which are normally applied to isolators, are also applied.

**Circuit segment of Figure 7.10(B):** This circuit segment is the circuit that is used to connect the two busbars. The busbars must be synchronized before connecting them. This circuit is also used to provide alternate route to connect a line to the busbar when the line-circuit breaker is out of service. The following interlock defines the operating constraints of this circuit.

I01: (C0(ON)& C0(FREE)) -----(7.4)

I02: (C0(ON)& C0(FREE)) -----(7.5)

These expressions ensure that the isolators are not operated if:

- C0 is closed.

In the event that the circuit breaker is stuck, the isolators would be allowed to operate as long as they satisfy either of these following conditions:

- Do not make or break currents.
- Have a short circuit path across the switch.

Note that SSP has an automatically embedded logic to check these constraints. It is, therefore, no need to write interlock constraints to include these statements.

**Circuit segment of Figure 7.10(C):** The purpose of the circuit breaker in this segment is to make or break the load currents and to interrupt the fault currents flowing from the busbars to the line Io. The purpose of the isolators is to select the bus to which the line is to be connected. The isolators are also used to disconnect the circuit breaker C from the rest of the network. An analysis of this circuit shows that the following interlock must be applied on this segment:

I41: (I42(ON)&I41(NOSCPATH)) | G41(ON) | G42(ON)  
| (C4(ON)& C4(FREE)) -----(7.6)

$$\begin{aligned} \text{I42: } & (\text{I41(ON)} \& \text{I42(NOSCPATH)}) \mid \text{G41(ON)} \mid \text{G42(ON)} \\ & \mid (\text{C4(ON)} \& \text{C4(FREE)}) \\ & ) \text{-----} (7.7) \end{aligned}$$

$$\begin{aligned} \text{I43: } & ( \\ & \text{G41(ON)} \mid \text{G42(ON)}) \mid \text{C4(ON HEALTHY)} \\ & ) \text{-----} (7.8) \end{aligned}$$

$$\begin{aligned} \text{G41: } & (\text{C4(FUNCTIONAL)} \mid \text{I41(ON)} \mid \text{I42(ON)} \mid \text{I43(ON)}) \\ & ) \text{-----} (7.9) \end{aligned}$$

$$\begin{aligned} \text{G42: } & (\text{C4(FUNCTIONAL)} \mid \text{I41(ON)} \mid \text{I42(ON)} \mid \text{I43(ON)}) \\ & ) \text{-----} (7.10) \end{aligned}$$

Equation 6 ensures that I41 must not operate when:

- I42 is already in its closed state but no short circuit path across I41 is established
- One of the two grounding switches is still closed
- The circuit breaker is closed but not stuck
- It makes or breaks currents (by default)

Equation 7 is similar to Equation 6 but is for switch I42.

Equation 8 tells that I43 must not operate when:

- Grounding switch G41 and circuit breaker C4 are closed
- Grounding switch G42 is closed
- The circuit breaker C4 is closed but not stuck
- It makes or breaks currents (by default).

As indicated by Equations 9 and 10, the grounding switches must not operate if

- Any of the isolators is closed, and
- C4 is not under repair.

**Circuit segment of Figure 7.10(D):** This is the circuit that plays two roles, one in transferring a line from one busbar to another and the other in bypassing a faulty circuit breaker. The circuit arrangement requires complex interlock constraints.

Normally, line Io is connected to the main busbar, Bus1, via I14, I13, C1 and I11. All other switches are normally open. An alternate arrangement is used when this state is not viable. This circuit arrangement can provide the connection between Io and the busbars via the following paths:

- Path 1: Io, I14, I13, C1, I11, Bus1
- Path 2: Io, I14, I13, C1, I12, Bus2
- Path 3: Io, I14, I15, Bus2, I02, C0, I01, Bus1
- Path 4: Io, I25, Nd, another-line's circuit breaker, Bus1.

These paths are listed in the order of preference. Path 1 must be used in normal operating conditions. Path 2 must be used as the alternate route if Bus1 has to be taken out of service. For Path 1, I11 is closed and I12 is open, whereas for Path 2, I12 is closed and I11 is open. The bus-coupler circuit is used during the transition from Path 1 to Path 2, and vice versa, without interrupting flow of power on the line. The following expressions provide the required interlocks.

$$\begin{aligned}
 I11: & \quad ( \\
 & \quad ( (I12(ON) \mid I15(ON) \mid I25(ON)) \& I11(NOSCPATH) ) \\
 & \quad \mid G11(ON) \mid G12(ON) \\
 & \quad \mid C1(ON \text{ FREE}) \\
 & \quad ) \text{-----} (7.11) \\
 I12: & \quad ( \\
 & \quad ( (I11(ON) \mid I15(ON) \mid I25(ON)) \& I12(NOSCPATH) ) \\
 & \quad \mid G11(ON) \mid G12(ON) \\
 & \quad \mid C1(ON \text{ FREE}) \\
 & \quad ) \text{-----} (7.12)
 \end{aligned}$$

Path 3 is used when the circuit-breaker C1 is out of service. This path uses the circuit breaker C0 to protect the line. The following interlock expression can be defined for the isolator I15.

```

I15:  (
      (C1(HEALTHY FUNCTIONAL)) |
      Bus2(DISABLED) |
      ( Bus2(ON) & I15(NOSCPATH) )
    )------(7.13)

```

If Io is out of service, C1 can be used to protect another circuit via I15 or I25. The following expression provides the required interlock for I25:

```

I25:  (
      I15(ON)|(C1(FUNCTIONAL)&Io(FUNCTIONAL)|I25(NOSCPATH))
    )------(7.14)

```

The following interlock constraints remain similar to the way they were in the case of the circuit of Figure 7.10(C).

```

I13:  (
      G11(ON)|G12(ON) |C1(ON HEALTHY)
    )------(7.15)

```

```

G11:  (C1(FUNCTIONAL)|I11(ON) | I12(ON) | I13(ON)
    )------(7.16)

```

```

G42:  (C1(FUNCTIONAL)| I11(ON) | I12(ON) | I13(ON)
    )------(7.17)

```

This approach was used for entering special interlock constraints of complex substations. There is no limit on the complexity or length of the interlock expressions that can be in the SSP. The constraints for Test Substations 6 and 7 were entered in this manner.

## 7.4. Testing Interlocking Schemes

Several tests were performed on the selected substations to determine the validity and suitability of the interlocking schemes created by the SSP. Conditions were created in which closing or opening of a switch would overload a line, violate the voltage limits on a

bus, cause generation level to be insufficient to supply the load, or ground a live node. A scheme was considered to be correct if it did not allow switching operations that would violate the operating constraints. Similarly, switches were selectively locked, or set as inoperable to verify if SSP would detect them and block the operation of those switches.

This section demonstrates how constraints are automatically applied in the operation of switches. The test results presented in this section concern Substations 1 and 6. The number of tests is limited to keep the size of this chapter manageable. These results include one or two categories selected from several scenarios. More results are provided in Appendix H in which a larger set of test cases on these and other substations are presented.

The first three tests were carried out in a double busbar substation identified earlier as Test Substation 1. It is assumed that the substation is working normally and all lines are connected to Bus 1. The bus coupler circuit breaker and the adjoining isolators are open as shown in Figure 7.1.

<b>Operating Command 1</b>	Open isolator 11. Part of the Substation and the dialog box are shown in Figure 7.11.
<b>Operating state</b>	Normal operating state of the substation.
<b>Result</b>	<p>The SSP refused to open isolator 11 and displayed the message shown in Figure 7.12. The reasons for not opening isolator 11 are as follows:</p> <ul style="list-style-type: none"> <li>• The interlocking constraint includes the requirement that isolator 11 should be allowed to open only if the adjacent circuit breaker (B1) is closed.. This condition was not satisfied in this case.</li> <li>• The isolator could have been closed if a short-circuit path existed across it; no such path existed in this case.</li> </ul>

<b>Operating Command 2</b>	Open circuit breaker B3. Part of the Substation and the dialog box are shown in Figure 7.13.
<b>Operating state</b>	Normal operating state of the substation.
<b>Result</b>	<p>The SSP did not close the circuit breaker. It showed the warning message that is shown in Figure 7.14. The reasons for the message being displayed are as follows.</p> <ul style="list-style-type: none"> <li>• The circuit breaker interrupts currents flowing to Load1.</li> <li>• The SSP needs confirmation from the operator that the interruption is deliberate. The operator must select “Yes” in the message box for completing the switching operation.</li> </ul>
<b>Operating Command 3</b>	Select “Yes” in the message box displayed in Figure 7.14 to confirm that the interruption of load was intended.
<b>Operating state</b>	The operating state was the one generated by Test 2.
<b>Result</b>	<p>The SSP closed circuit breaker B3 as shown in Figure 7.15. The reasons for opening the circuit breaker are as follows:</p> <ul style="list-style-type: none"> <li>• It does not cause overloading of the remaining lines.</li> <li>• It does not cause over voltage or undervoltage at any bus.</li> </ul>

These tests and a number of other tests presented in Appendix H demonstrate that SSP correctly identifies safe switching operations and unsafe switching operations in configurations of substations where the interlocks were automatically generated by the SSP. The switching scheme permits only those switching operations that are safe.

The fourth test result presents a case that is similar to cases 1 to 3 except that the configuration of the selected substation is more complex. Substation 6 was selected for these tests. Extra interlocking constraints were defined for this substation in addition to those automatically generated by the SSP

For this test, the normal operating state is when all circuits are connected to Bus1, as is

shown in Figure 7.6. Interlock constraints that are specific to this substation were kept in a text file. These constraints are loaded by selecting, “Edit” and “ReadLock” from the menu before issuing the command to compile the substation data.

<b>Operating Command 4</b>	Close grounding switch G41 in the substation shown in Figure 7.16.
<b>Operating state</b>	Normal operating state of the substation.
<b>Result</b>	<p>The SSP refused to close G41 and displayed the message shown in Figure 7.17. The reasons for not permitting this operation are as follows:</p> <ul style="list-style-type: none"><li>• Circuit breaker C4 is not in repair mode</li><li>• Isolators I41, I42 and I43 are closed; these isolators must be open before the grounding switch is allowed to be closed as defined in the interlock expression entered for G41.</li></ul> <p>Unlike in previous test cases, the SSP checked the user-defined operating constraints instead of the violation of the default operating constraints. The SSP would have checked the default interlock constraints if the user-defined interlock expression had not blocked the switching operation. The operation could proceed only if both the used-defined and default logic determined that the operation is permissible.</p>

These examples, and examples presented in Appendix H, demonstrate that the technique for developing interlocking schemes described in this thesis works well. They also demonstrate that the interlocking schemes can be developed for any substation using the approach presented in this thesis. Desired degree of flexibility and restraints can be provided while ensuring safety of the substation equipment and personnel working in those substations.



## **7.5. Switching Sequence Test**

Tests were performed on the SSP by issuing commands that require it to operate multiple switches in an appropriate order. When such commands are entered, the CHU of the SSP interpreted the commands and determined the appropriate sequence. Random events, also referred to as triggering events, were introduced while substations were running under normal operating conditions. The SSP initiated a series of switching operations that took the circuit to an operating state appropriate to the command issued to it or the event encountered by it. Samples of test results are presented in this section to demonstrate the important sequence-switching features of the SSP.

When presenting the results of earlier tests, the events recorded by the SSP in its “event log” were not introduced. When commands were issued to operate only one switch, the need to observe the order of operation was not great. In the following tests, the order of operation is important and, therefore, the events recorded by the “event log” of SSP are shown with the test results.

### **7.5.1. Connecting and disconnecting lines**

Connecting a line to a busbar or disconnecting a line from a busbar is relatively straightforward. The SSP allows to connect a circuit to a busbar provided a valid power flow path exists between the circuit and the busbar. The SSP also allows a circuit to be disconnected from the busbars.

<b>Operating Command 5</b>	Connect 7L4 to busbar in the substation shown in Figure 7.18.
<b>Operating state</b>	The line 7L4 is disconnected from the busbars, as shown in Figure 7.18.
<b>Result</b>	The SSP connected line 7L4 to Bus1 by closing the following switches:

Isolator I44

Isolator I43

Isolator I41

Circuit breaker C4

The final state of the switches are shown in Figure 7.19, which also shows the event log that records the order of operations.

The circuit breaker has been closed after the isolator to make sure that no current flows through the isolators while they are closing.

The following points are to be noted in this operation:

- The SSP connected the line to Bus1, which was already energized, through the most preferable path.
- The SSP closed the isolators involved in the path before closing the circuit breaker.
- The SSP closed the switches starting from source end and progressing systematically towards the bus.

The SSP continuously logs the switching activities in a memory buffer but does not write the information in a file. The SSP can, however, be forced to write the information in the log file by selecting “File” and “Save Event Log” from the menu. In these cases, the SSP was instructed to write the event log shown in Figure 7.19.

### **7.5.2. Transferring busbars**

When a busbar has to be taken out of service, all the lines connected to that bus have to be connected via an alternate bus. The SSP evaluates the possibility of establishing alternate connections and establishes the connections through alternate routes before disconnecting a busbar from the rest of the system.

<b>Operating Command 6</b>	Transfer lines from busbar Bus1 to Bus 2 in the substation shown in Figure 7.20.
<b>Operating state</b>	Normal. All lines are connected to Bus1.
<b>Result</b>	The SSP transferred lines from Bus1 to Bus2 resulting in the final configuration shown in Figure 7.21. The SSP closed the following switches:

Bus coupler isolator I01 and I02

Bus coupler circuit breaker C0

Bus selector switches I42, I52, I12, I22 and I32

The SSP then opened the following switches:

Bus selector switches I11, I21, I31, I41 and I51

Bus coupler circuit breaker C0

Bus coupler isolators I01 and I02.

The switching operations carried for transferring the lines from Bus1 to Bus2 and the final operating state of the substation are shown in Figure 7.21. During this bus transfer process the SSP performed the following operations:

- Energized Bus2 through bus coupler circuit and brought Bus1 and Bus2 to the same voltage level.
- Created a short circuit path across the bus selector switched.
- Connected all the lines to Bus2 via the bus selector switches connected to Bus2.
- Disconnected lines from Bus1 by opening bus selector switches adjacent to Bus1.
- Disconnected Bus1 from Bus2 by opening the bus coupler circuit.

### **7.5.3. Isolating switches**

In order to isolate a component, all paths radiating from that component must be opened and at least one isolator must be opened along each radiating path. The following tests were

conducted to show that the SSP ensures that such conditions are achieved and power flow through the component is maintained through an alternate route.

<b>Operating Command 7</b>	Isolate the circuit breaker C4 in the substation shown in Figure 7.22.
<b>Operating state</b>	Normal except that circuit breaker C4 needed for maintenance or repair.
<b>Result</b>	<p>The SSP isolated the circuit breaker C4. The resulting state of the substation and the switches operated during this process are listed in Figure 7.23. The SSP closed the following switches:</p> <ul style="list-style-type: none"><li>Bus coupler isolators I01 and I02</li><li>Bus coupler circuit breaker C0</li><li>Circuit by-pass isolator I46</li></ul> <p>The SSP opened the following switches:</p> <ul style="list-style-type: none"><li>Circuit breaker C4</li><li>Isolators I41 and I43</li></ul> <p>During this process an alternate power flow path has been established before opening the branch containing the circuit breaker B4.</p>

#### **7.5.4. Isolating busbars**

The process of isolating a busbar is similar to isolating a switch except that a busbar serves several lines while a switch controls one circuit only. As was done in other cases, an attempt is made to maintain the power flow through alternate routes if at all possible.

<b>Operating Command 8</b>	Isolate Bus1 in the substation shown in Figure 7.24.
<b>Operating state</b>	Normal.
<b>Result</b>	The SSP isolated Bus1. The lines connected to Bus1 were first connected to Bus2 by closing the following switches:

Bus coupler switches 1, 2 and B5

Bus selector isolators 4, 6, 10, 8

Bus1 was then isolated by opening the following switches:

Bus selectors 3, 5, 7, 9

Bus coupler switches B5, 2, and 1

The switching operations performed and the final state of the substation are shown in Figure 7.25.

#### **7.5.5. Bypassing faulty equipment**

Sometimes a component may not be isolated immediately but rather be bypassed from the power flow paths. Such bypassing may not isolate the component. Even after bypassing a component from the power flow path, a component must be isolated before the personnel perform any work on the component.

#### **Operating Command 9**

A. Instruct SSP to bypass circuit breaker C4 in the substation shown in Figure 7.26.

B. Connect line 7L4 to busbar in the substation shown in Figure 7.27.

#### **Operating state**

Line 7L4 is disconnected from busbars. All other lines are connected to Bus1. Circuit breaker C4 is not healthy.

#### **Result**

A. The SSP placed circuit breaker C4 in “bypass” mode to exclude it from new power flow paths. This state change was captured by the event log shown in Figure 7.26.

B. The SSP connected line 7L4 to Bus1 while bypassing circuit breaker C4 by closing the following switches:

Isolators I44, I46

Bus coupler switches I02, I01 and C0

The final state of the substation and the switches operated during the transition are shown in Figure 7.28.

The effect of operating a switch on bus voltages and line currents of the system are evaluated by performing load flow calculations. Section H.3 describes how the SSP facilitates the incorporation of the load flow in the validation of switching operations. It also describes the graph plotting module of SSP that plots line currents and bus voltages for graphical comparison of their values.

## **7.6. Summary**

The concepts formulated for the computer-aided design of substation switching schemes were successfully implemented in a software application package called SSP. The developed concepts were validated by designing switching schemes for eight substations. The results show that the techniques presented in this thesis are viable and can be implemented in an appropriate software package.

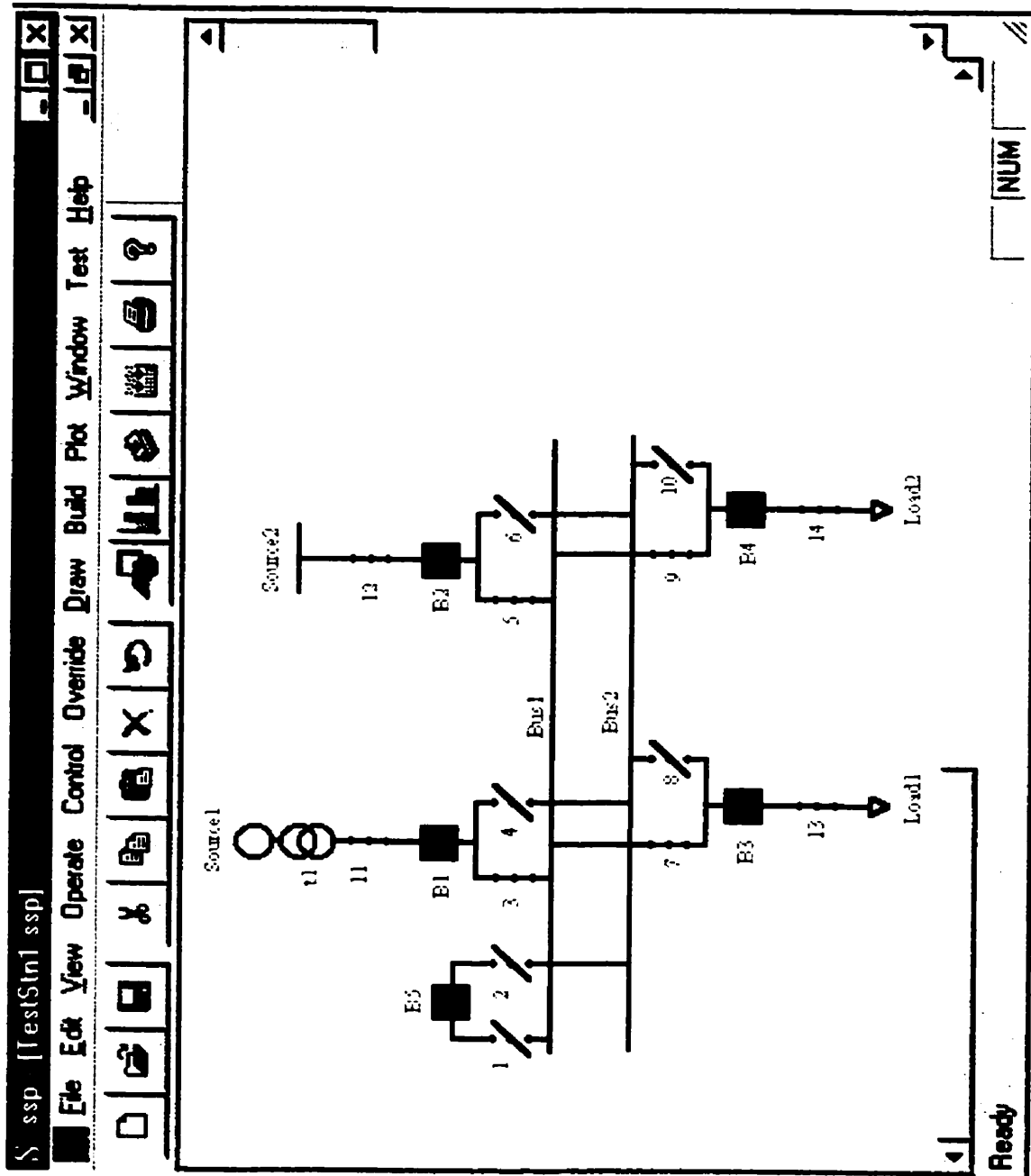


Figure 7.1. A section of a double busbar substation.

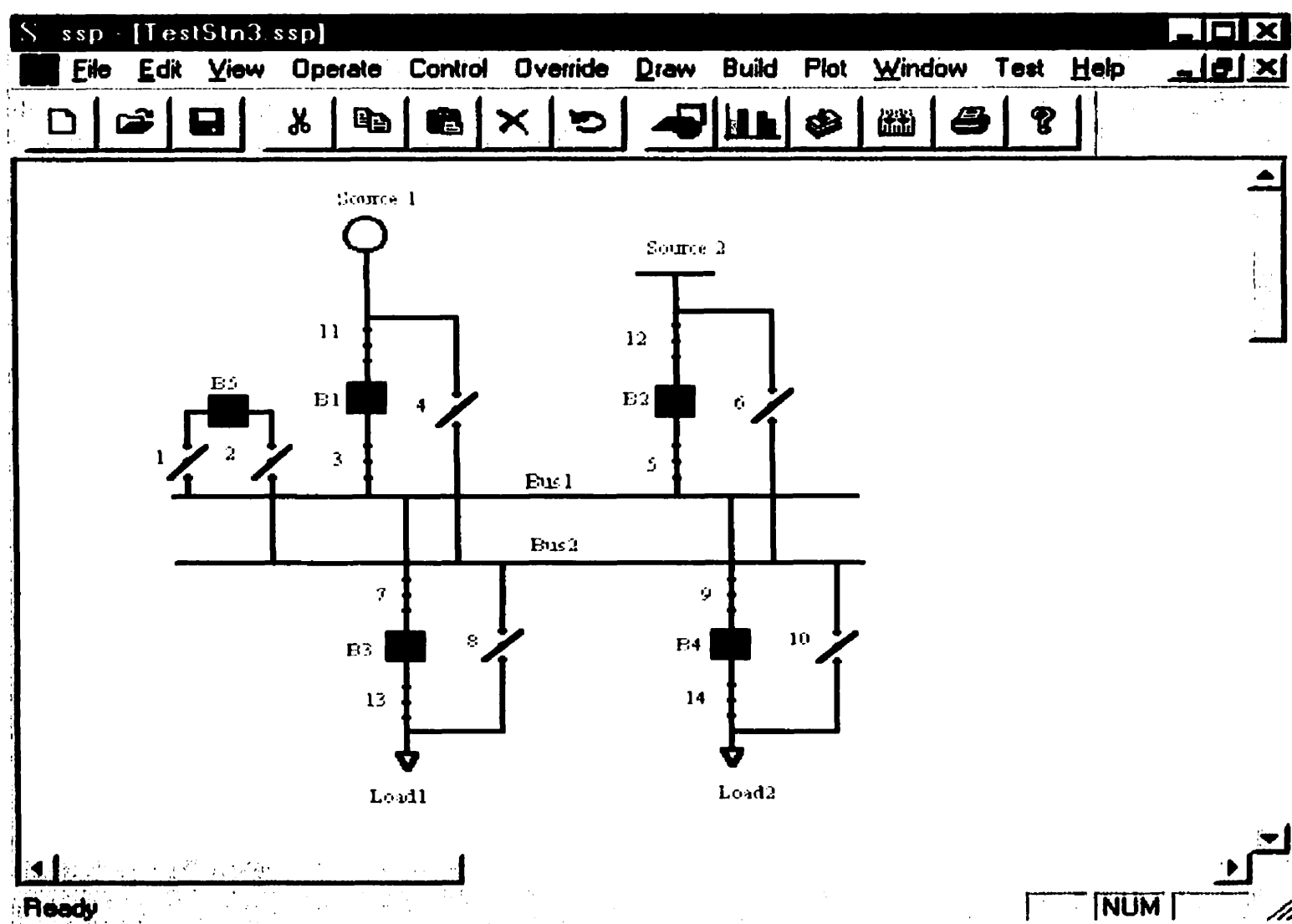
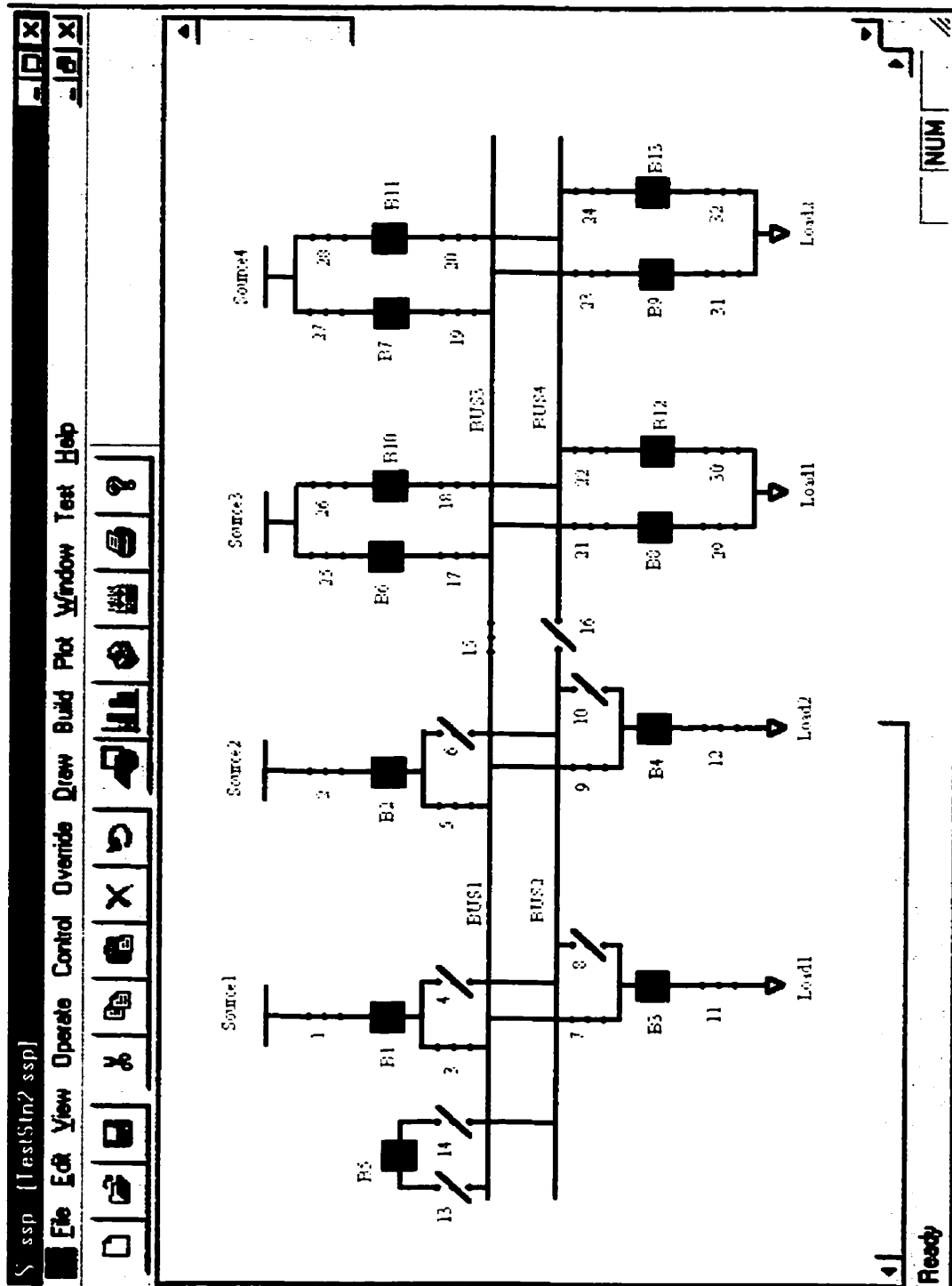
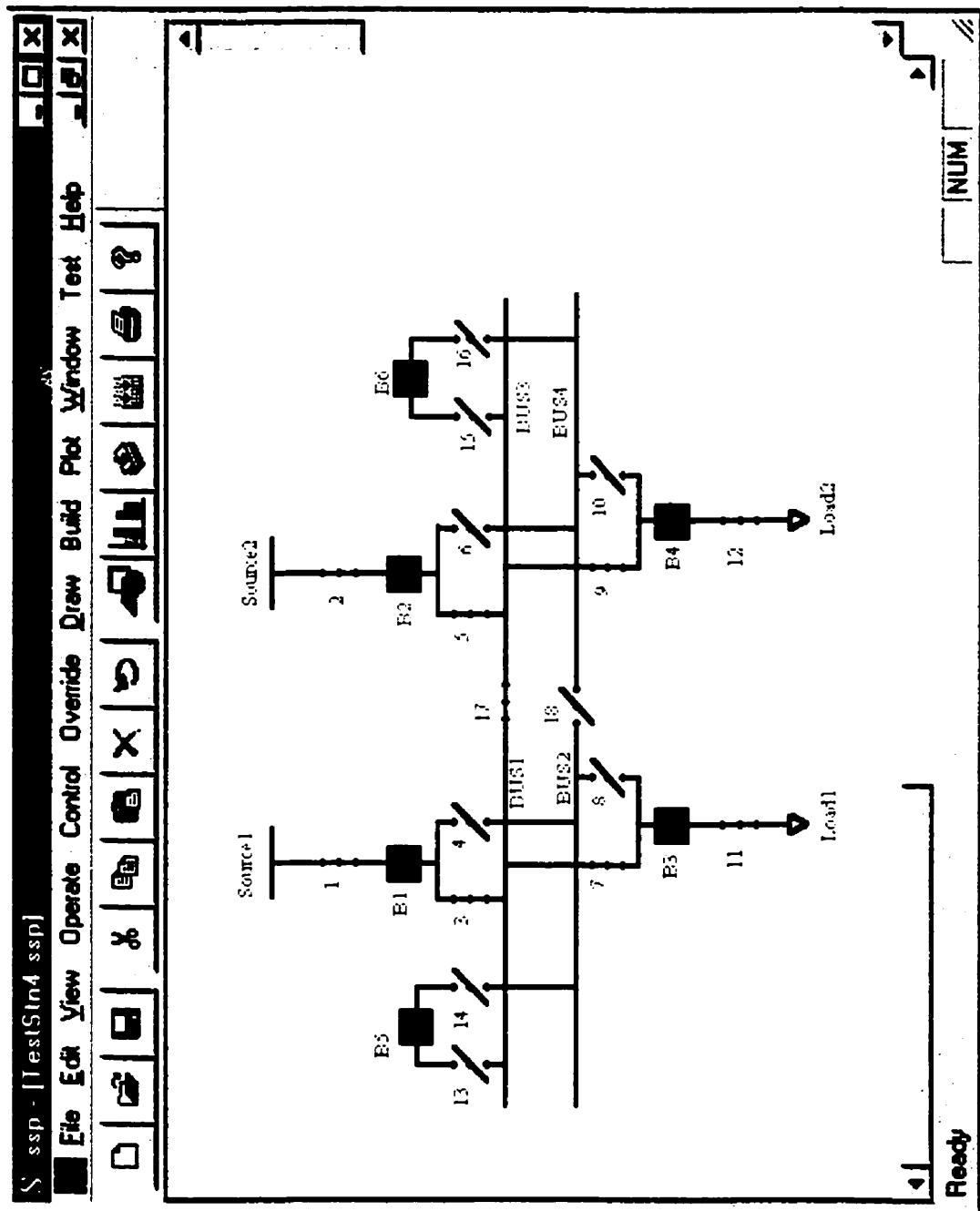


Figure 7.2. A section of a main-and-transfer busbar substation.

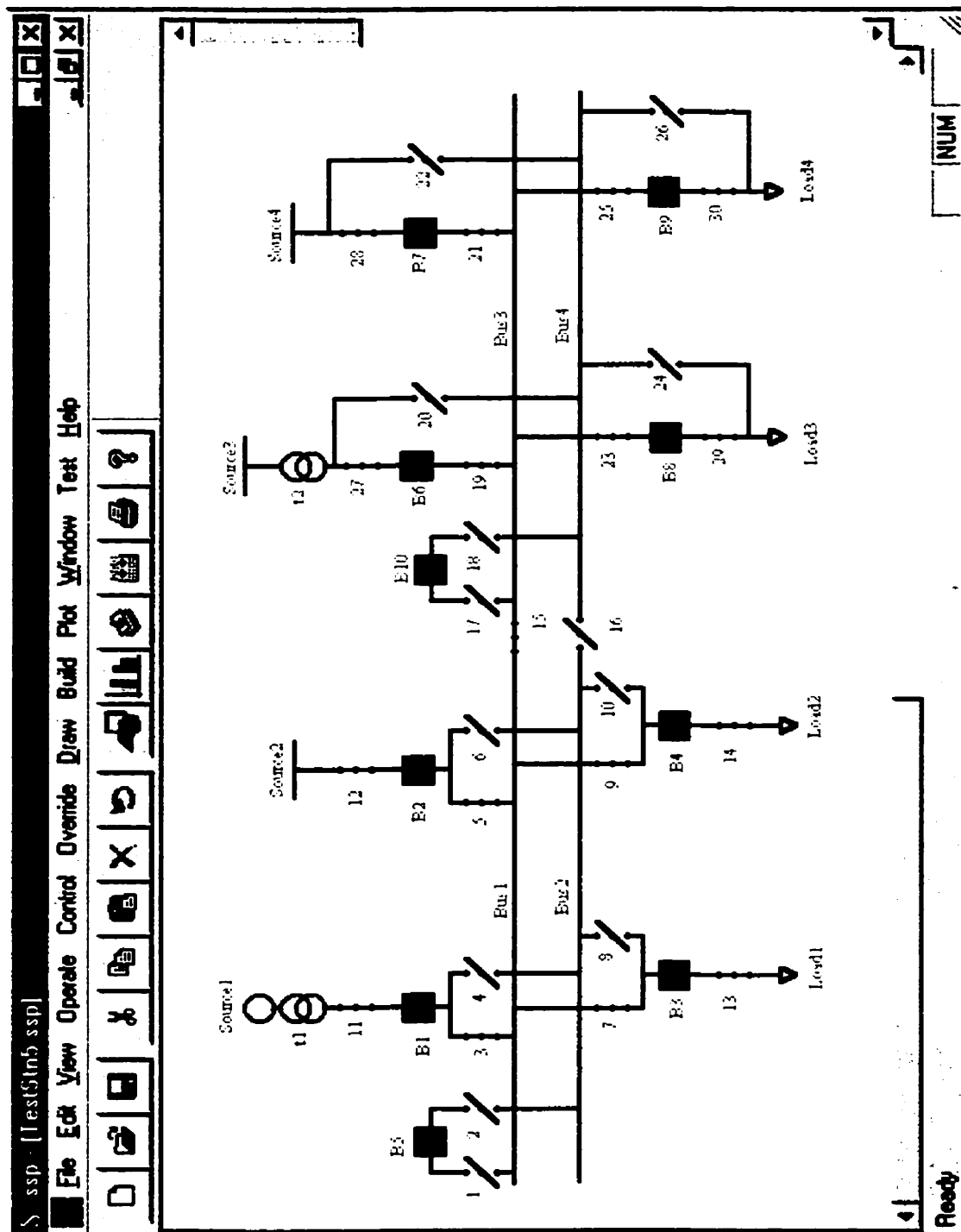




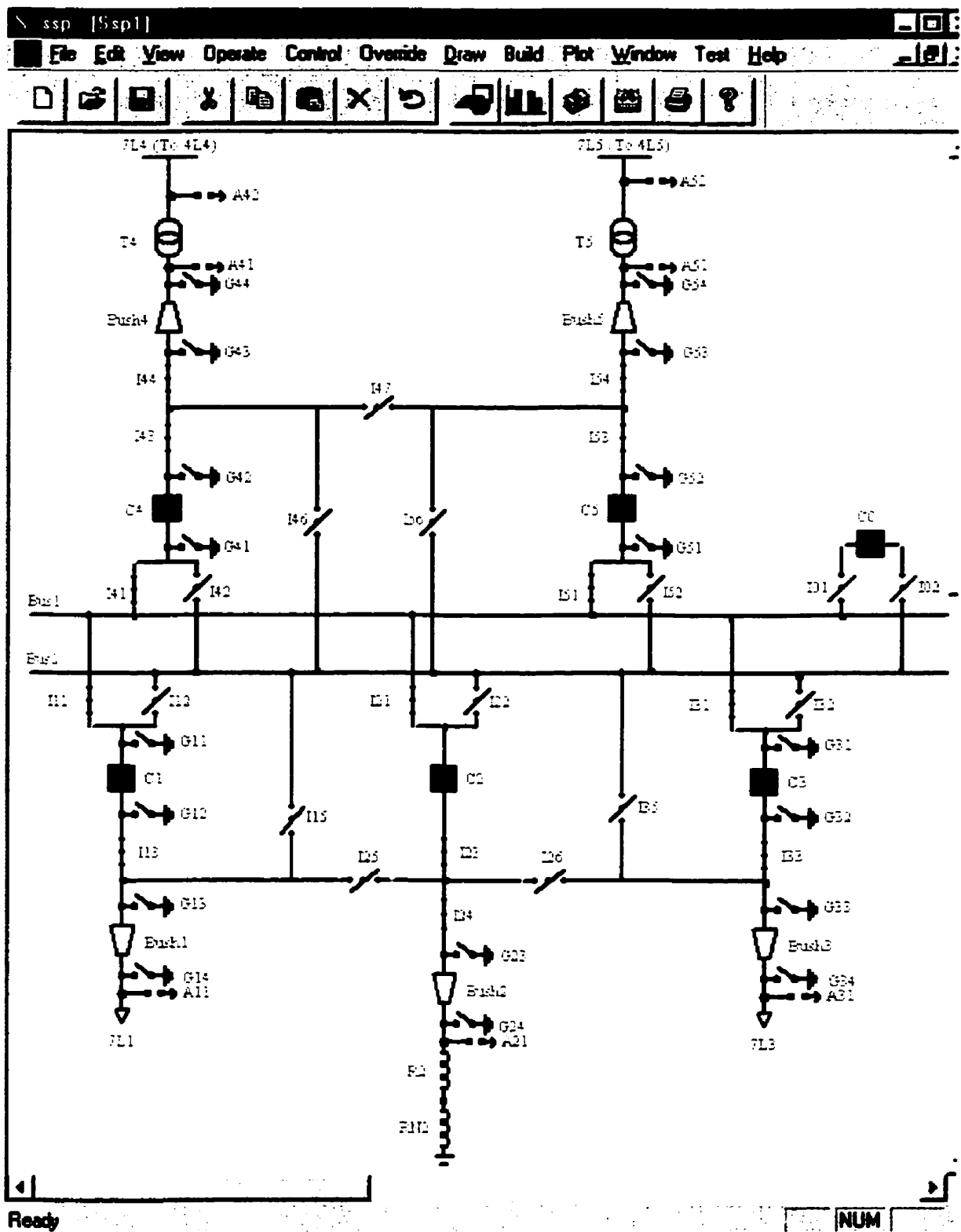
**Figure 7.3.** A combination of double busbar and two circuit-breaker sections.



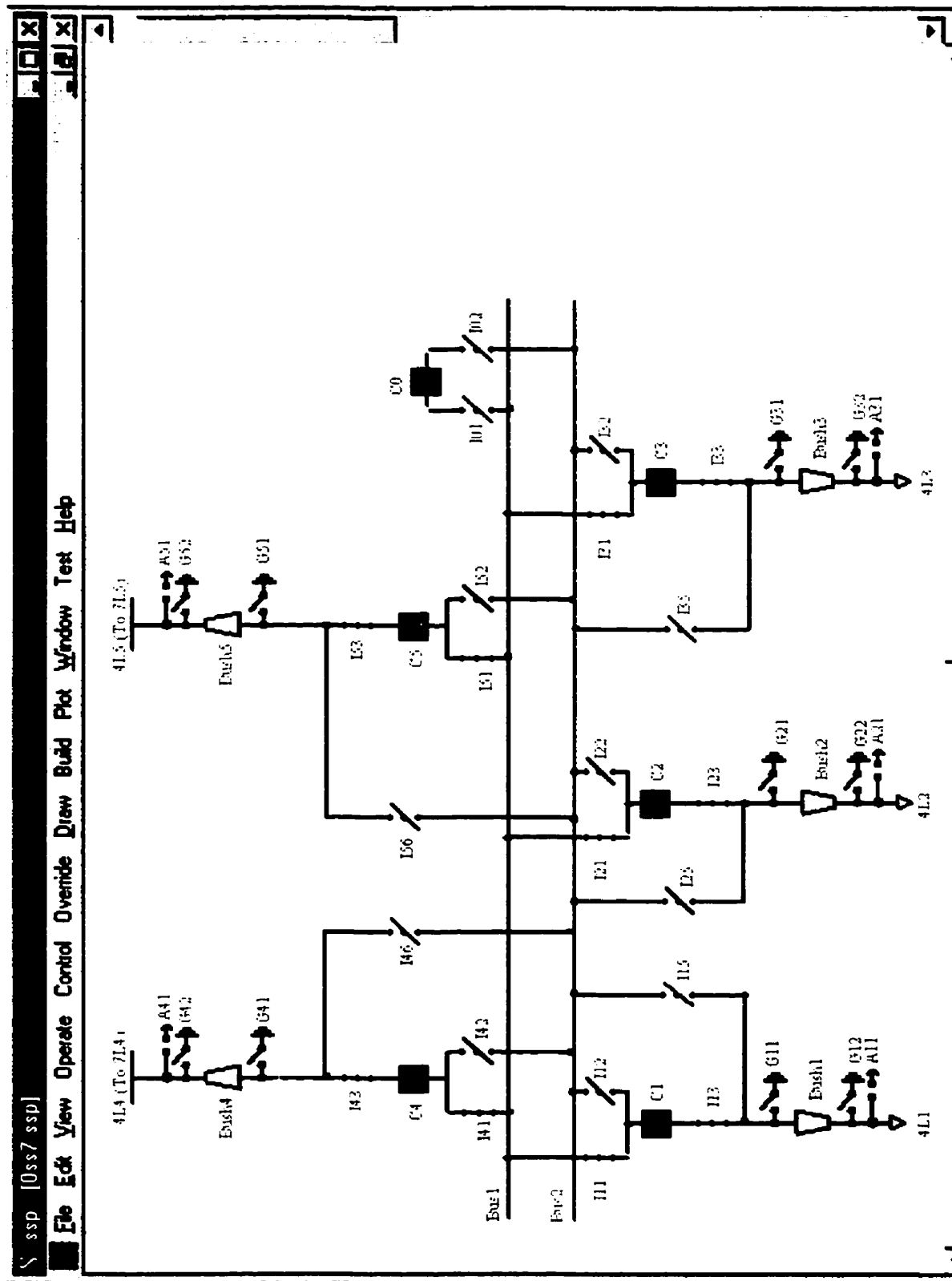
**Figure 7.4.** A prt of a sectionalized double busbar substation.



**Figure 7.5.** A substation with a combination of double busbar and main-and-transfer busbar sections.



**Figure 7.6.** A 765kV double busbar substation with bypass isolators.



**Figure 7.7.** A 400kV double-busbar substation with by-pass isolators.

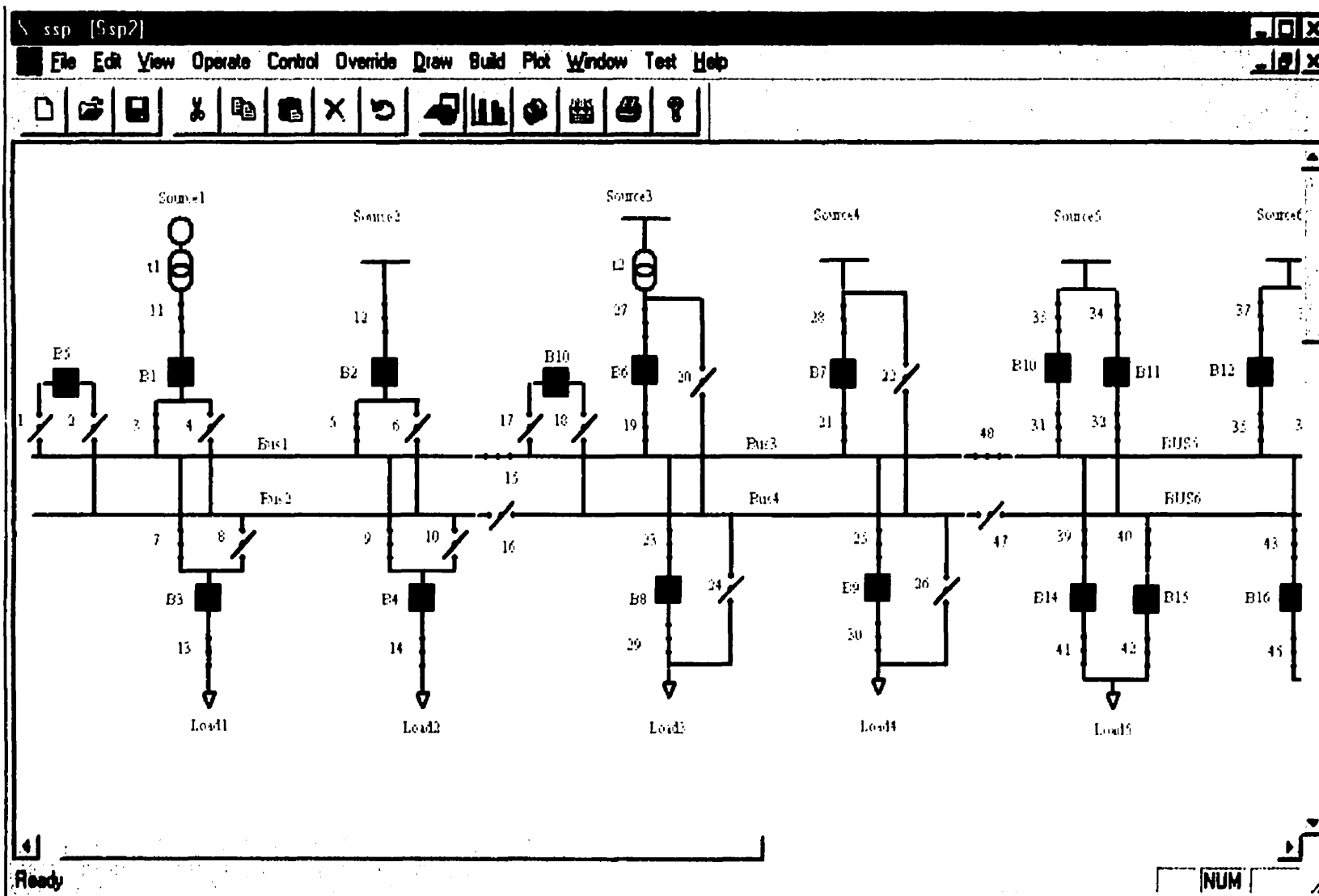


Figure 7.8. A three section substation.

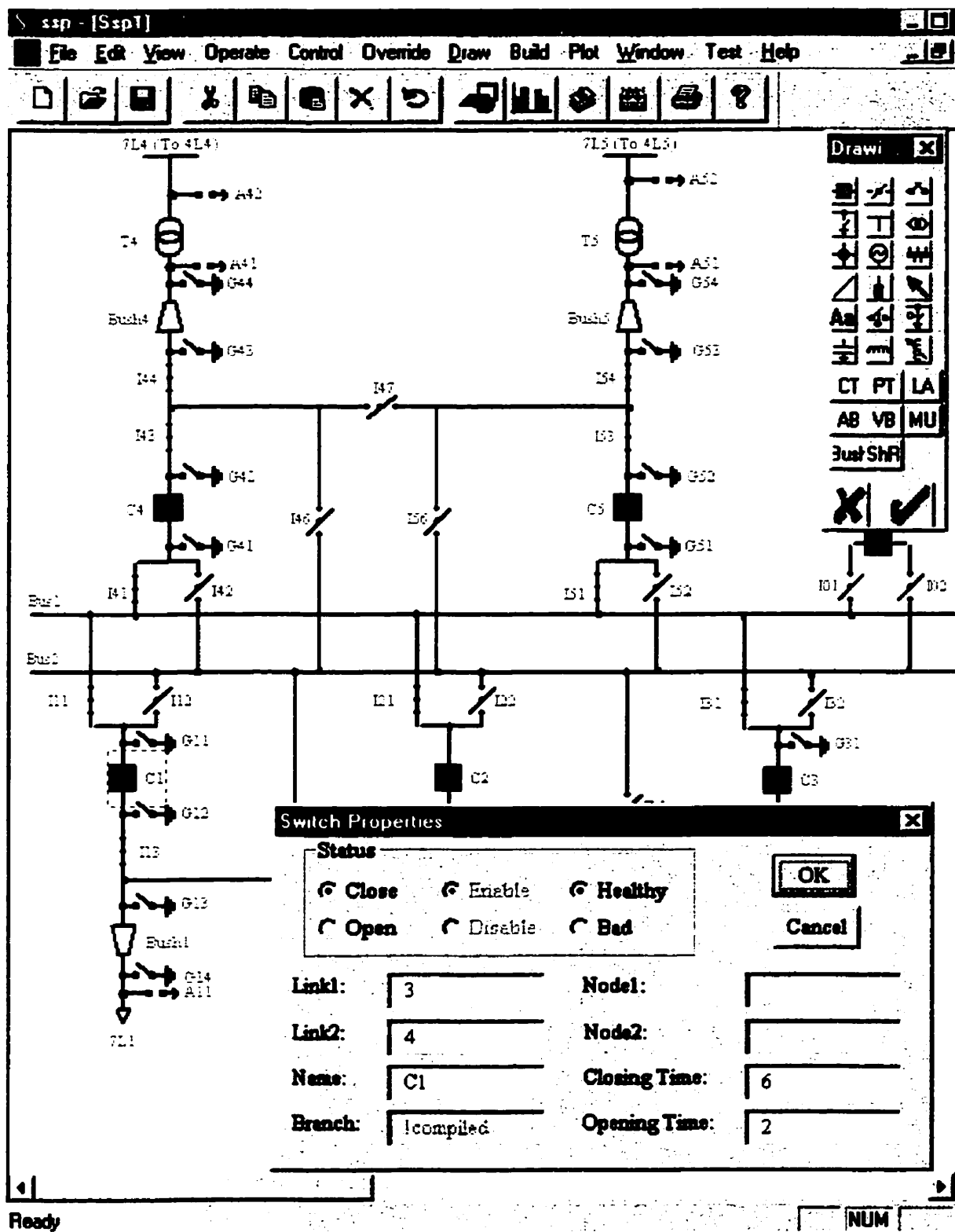
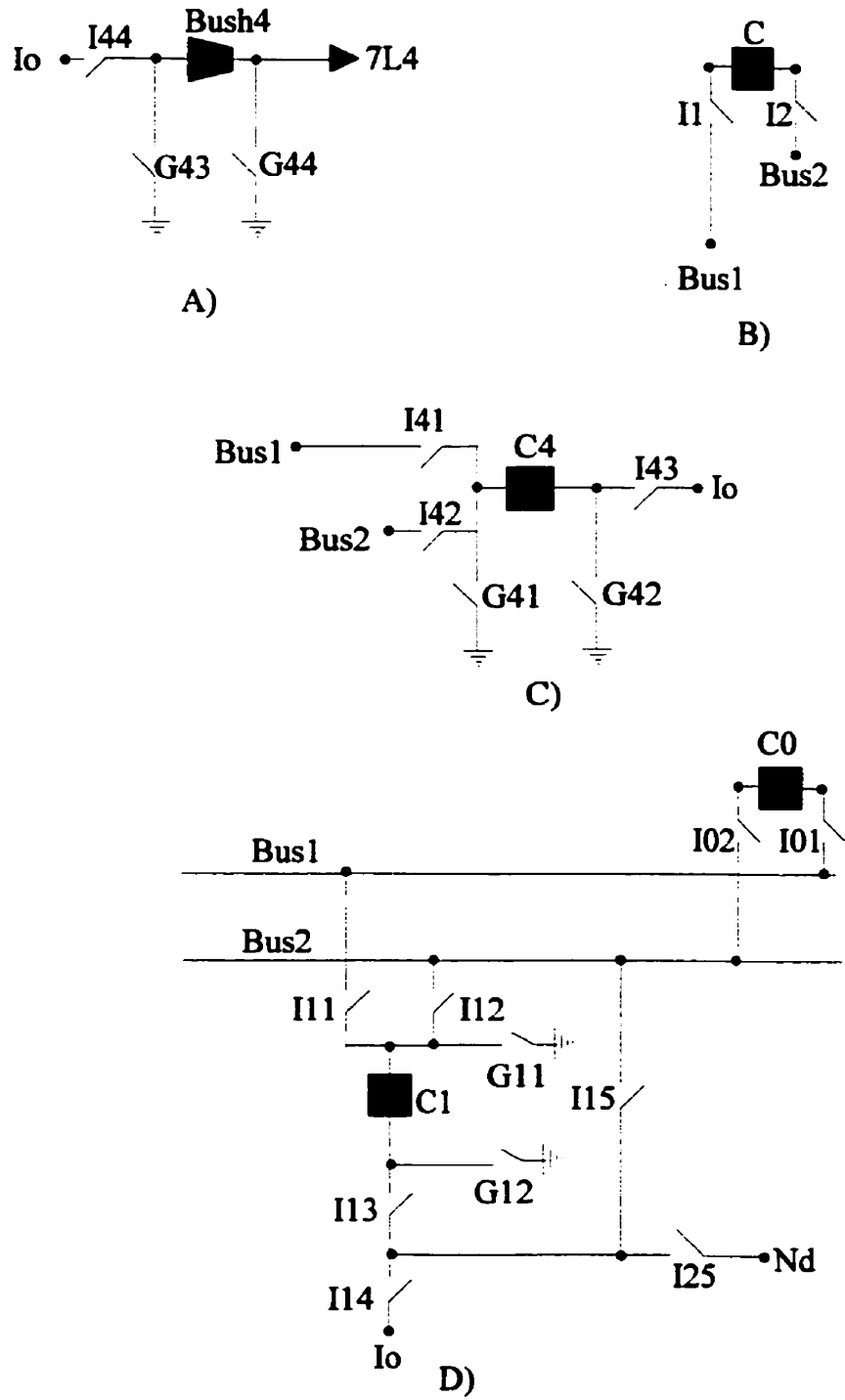


Figure 7.9. Data being entered for a switch in the GUI environment of the SSP.



**Figure 7.10.** Circuit segment patterns that build Test Substation 6.



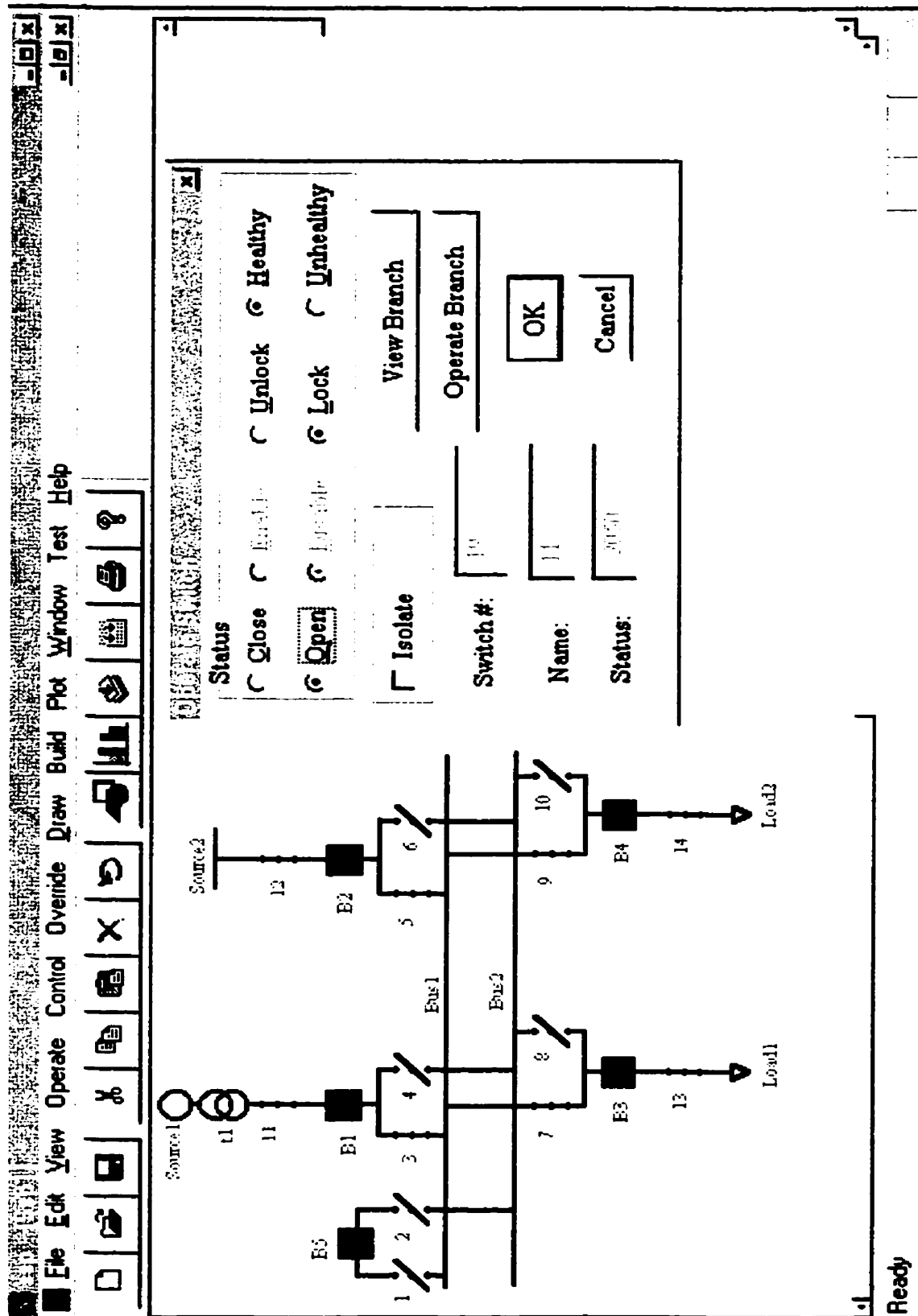
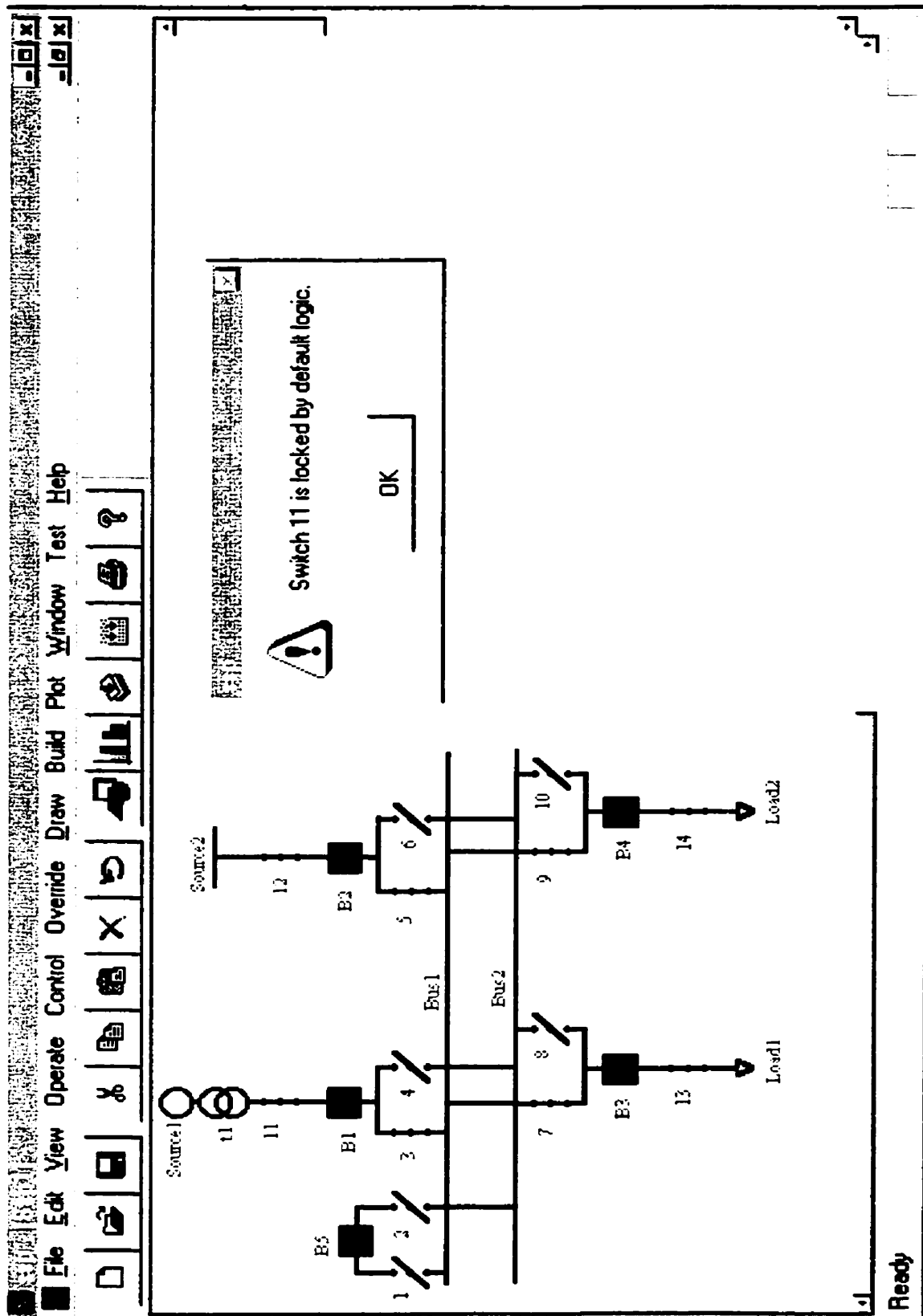


Figure 7.11. Command being issued to open isolator 11.



**Figure 7.12.** The SSP refuses to operate isolator 11.

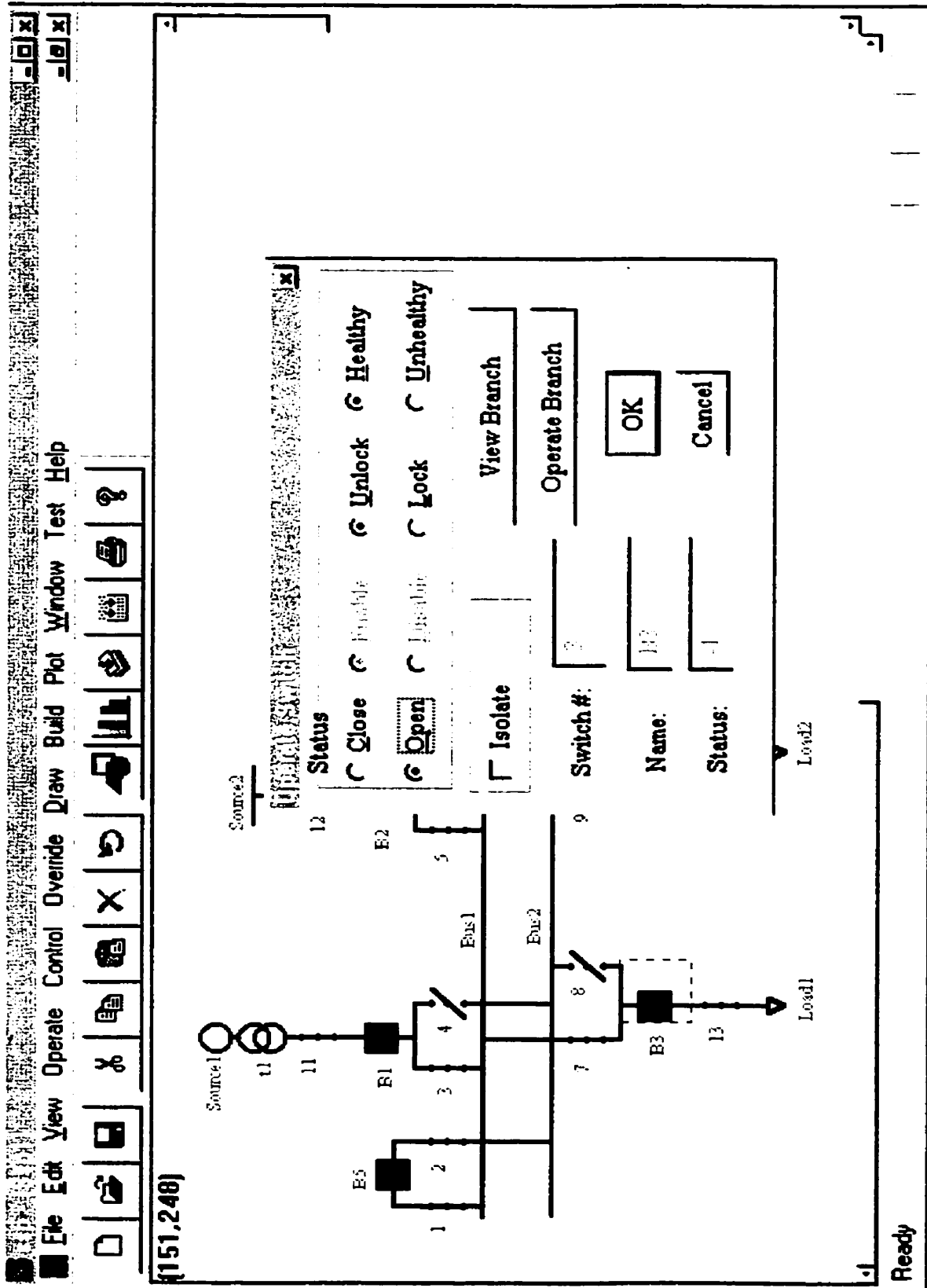


Figure 7.13. Command being issued to open circuit breaker B3.

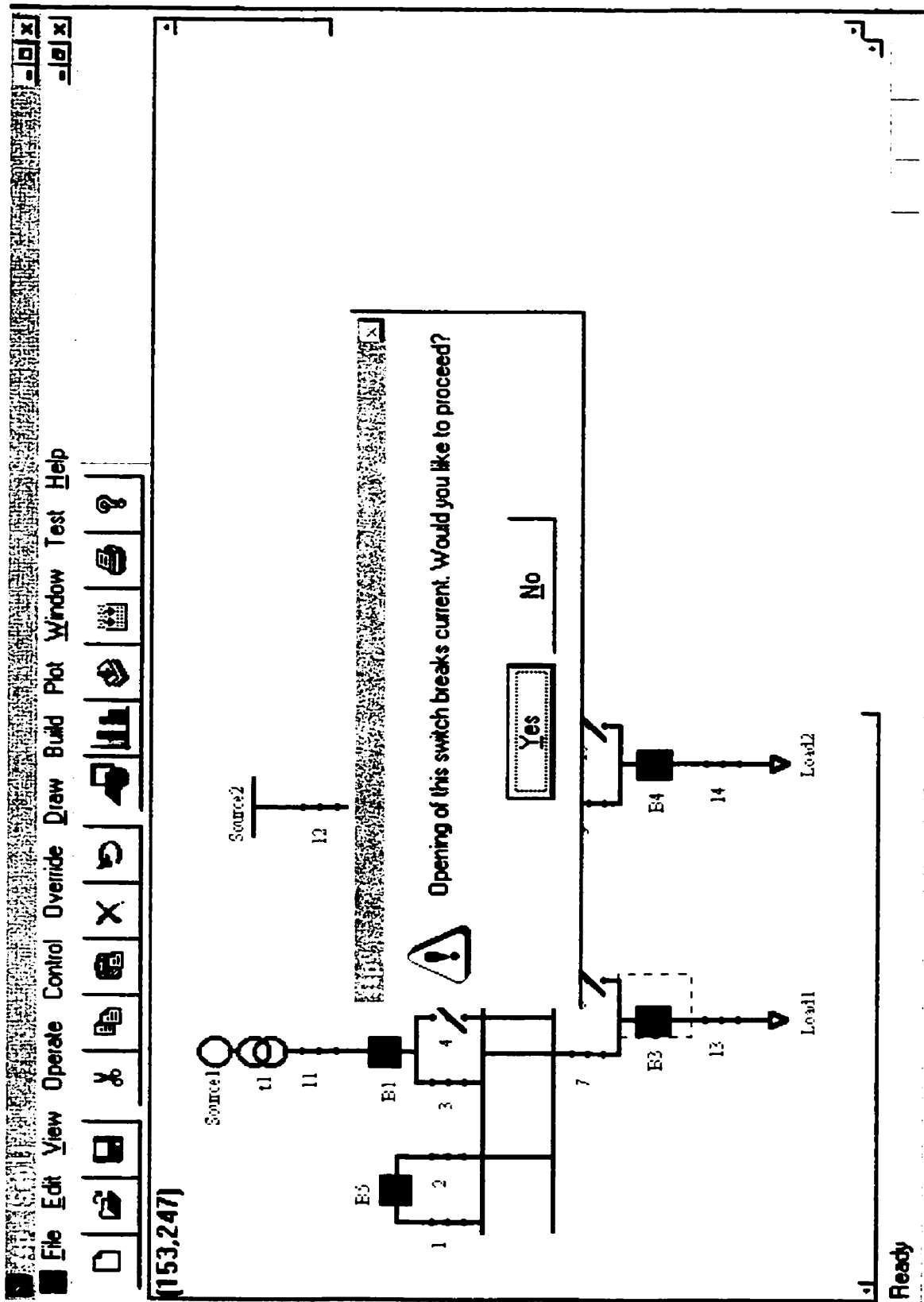


Figure 7.14. The SSP warns about the consequences and asks the operator to confirm the operation.

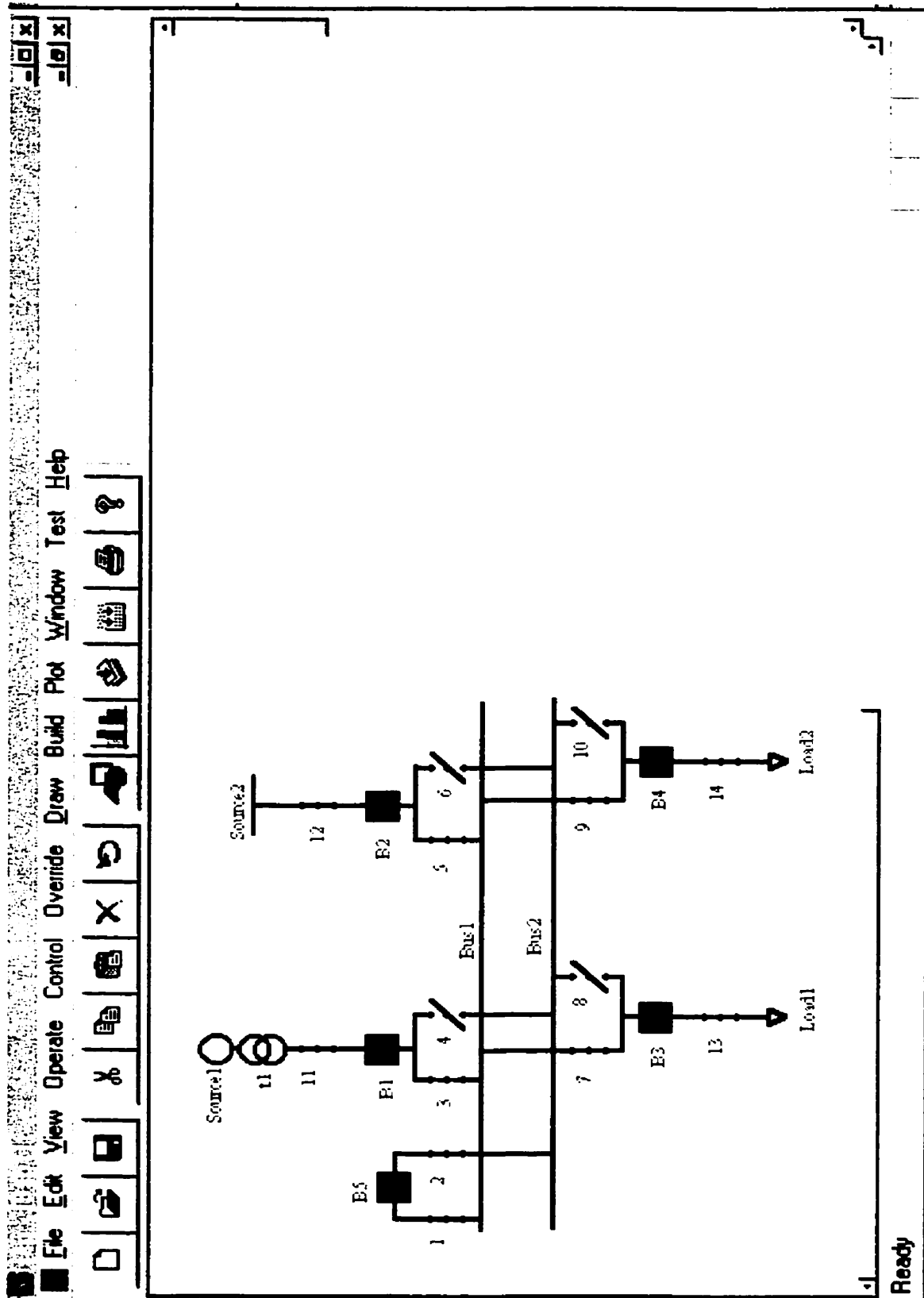


Figure 7.15. The SSP opens circuit breaker B3.

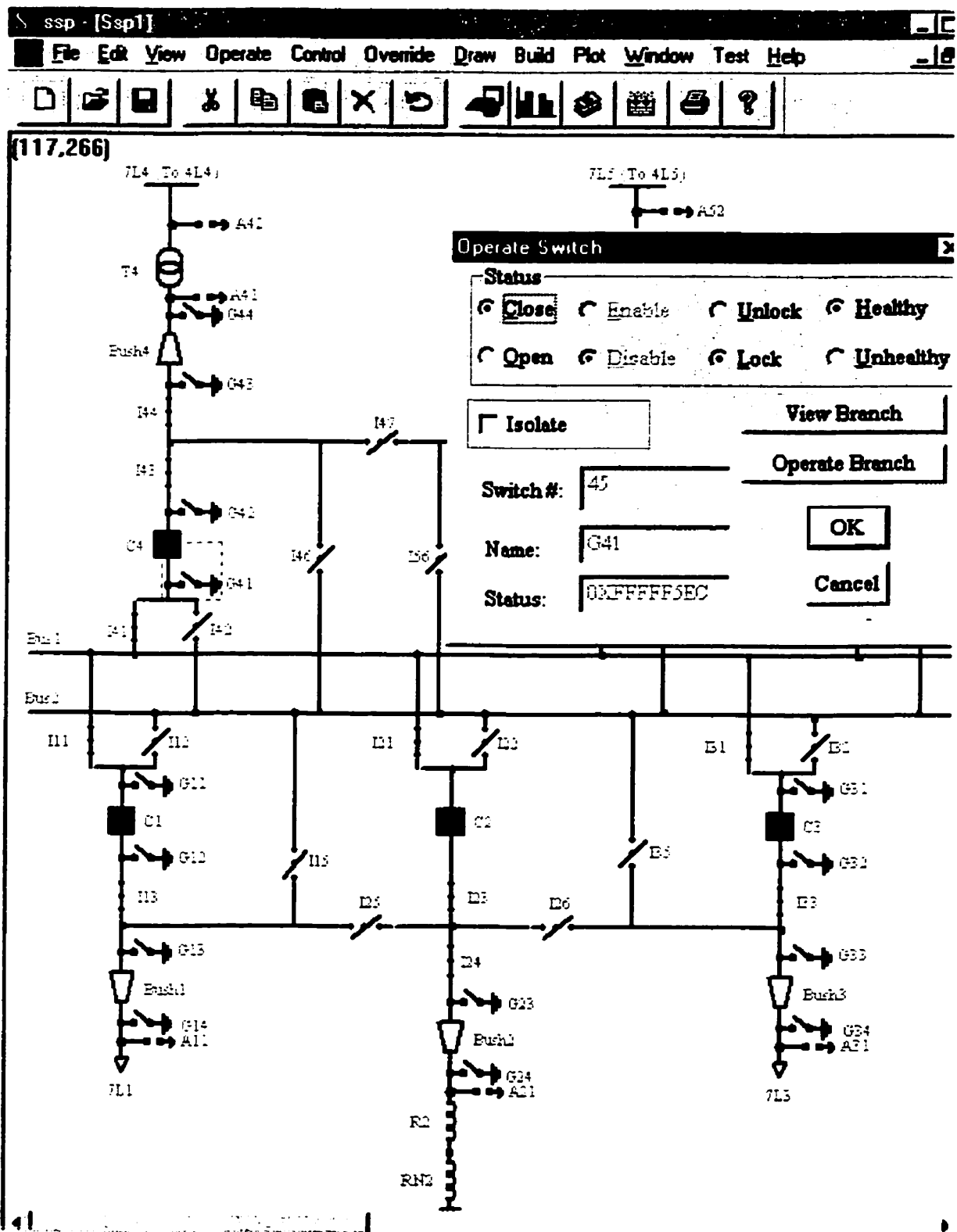
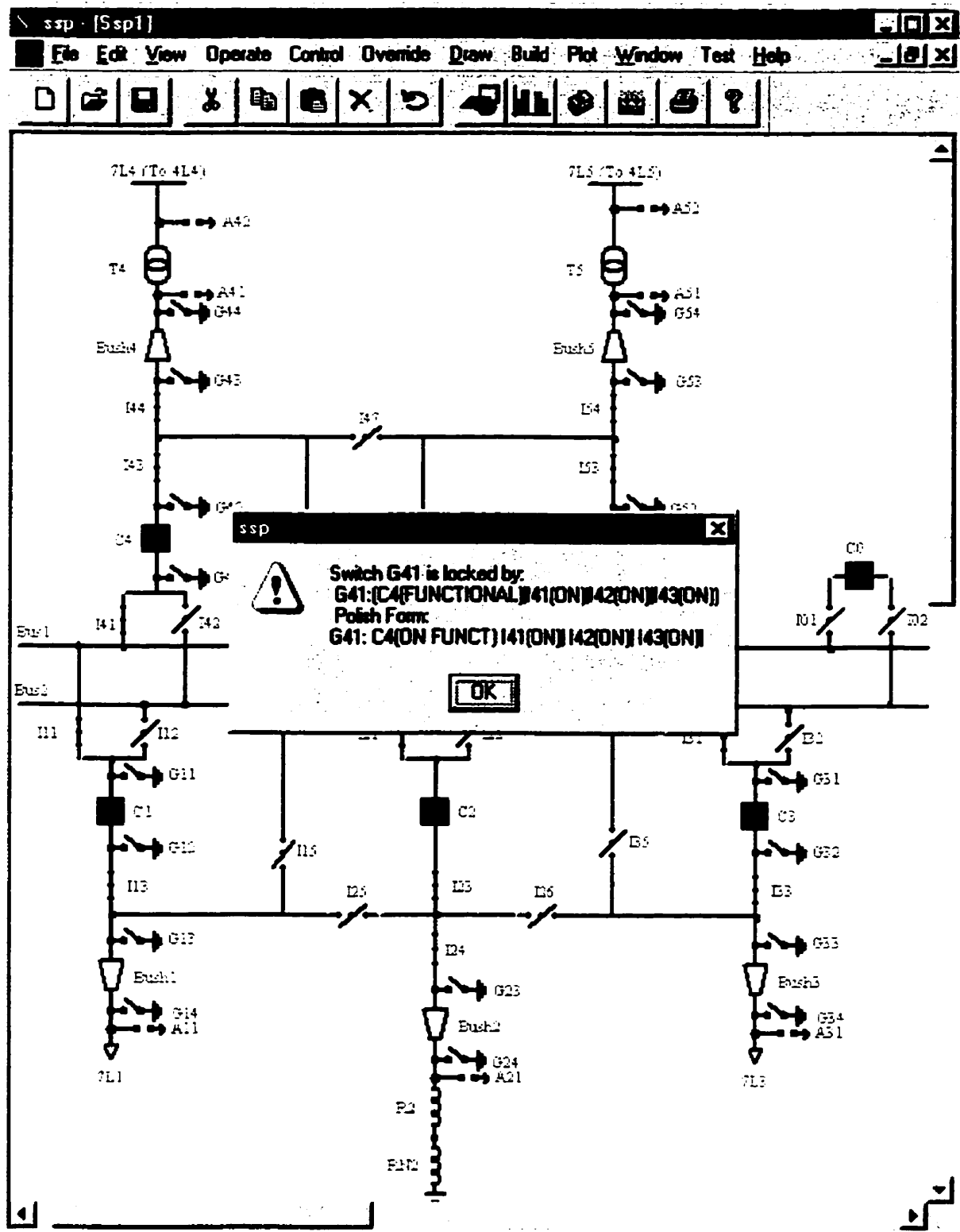


Figure 7.16. Command being issued to close grounding switch G41.



**Figure 7.17.** The SSP refuses to close G41 and displays the interlock constraint that blocked the operation.

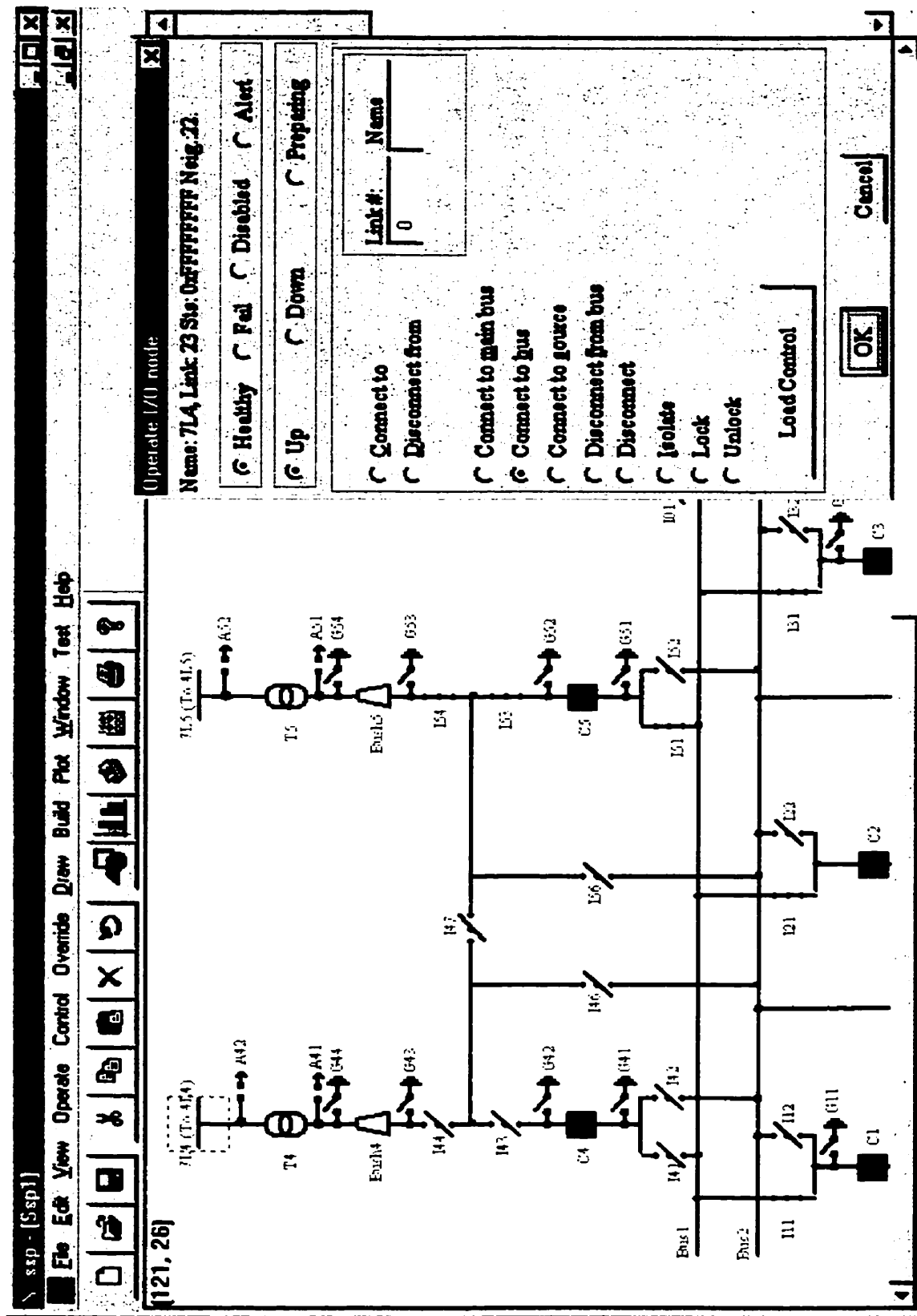
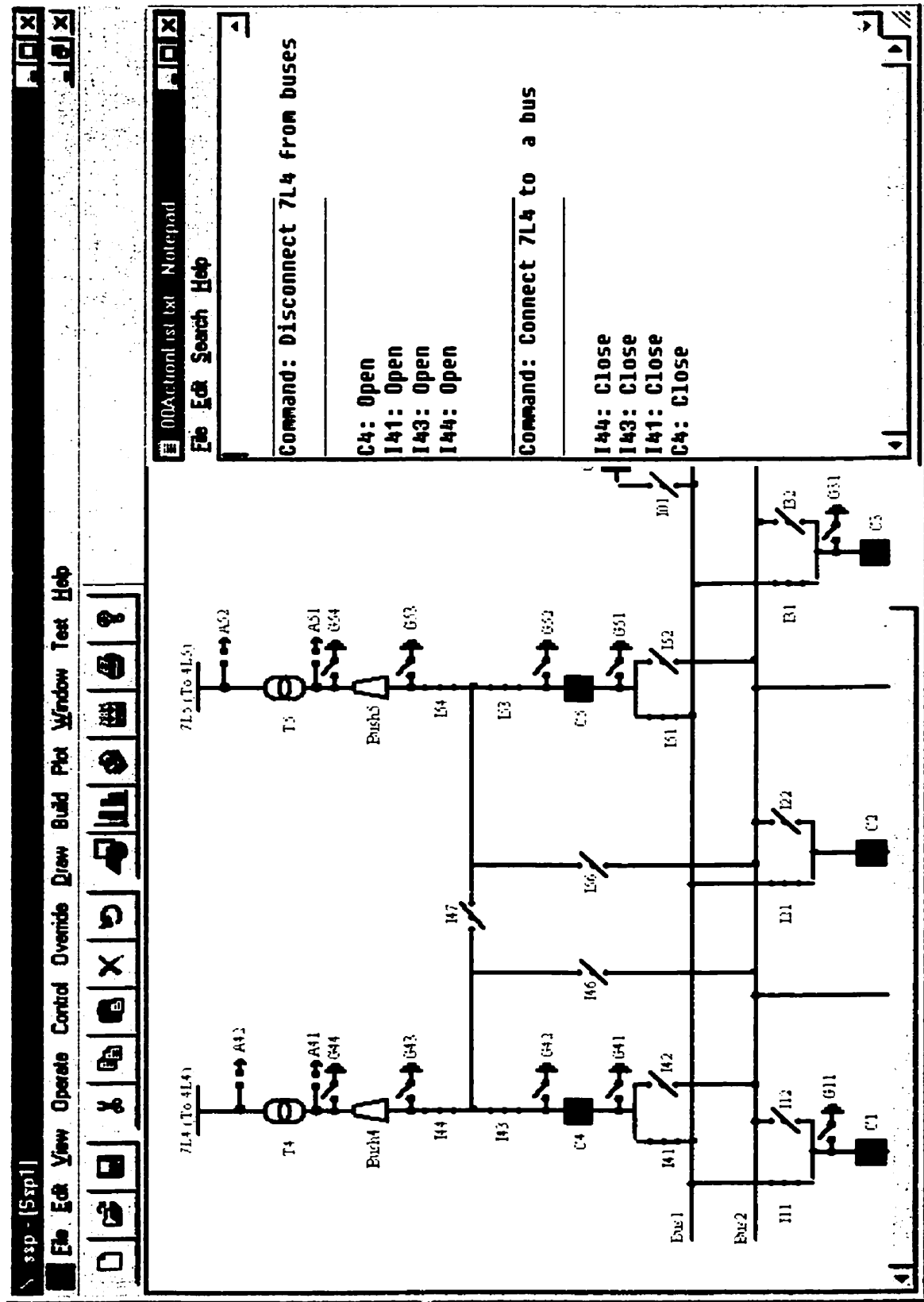


Figure 7.18: Command being issued to connect line 7L4 to a busbar.





**Figure 7.19:** The status of the switches after the SSP connected line 7L4 to Bus1.

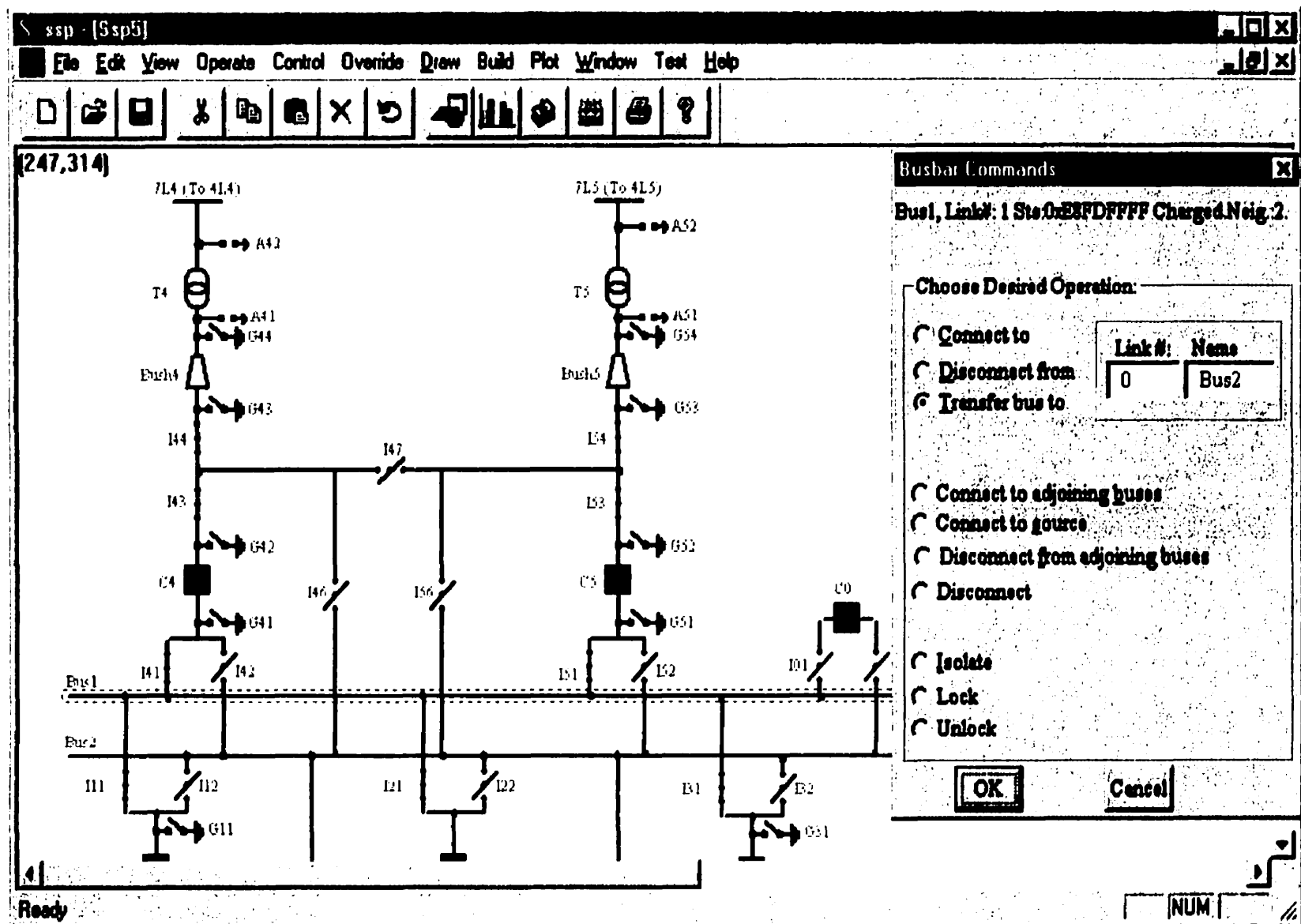
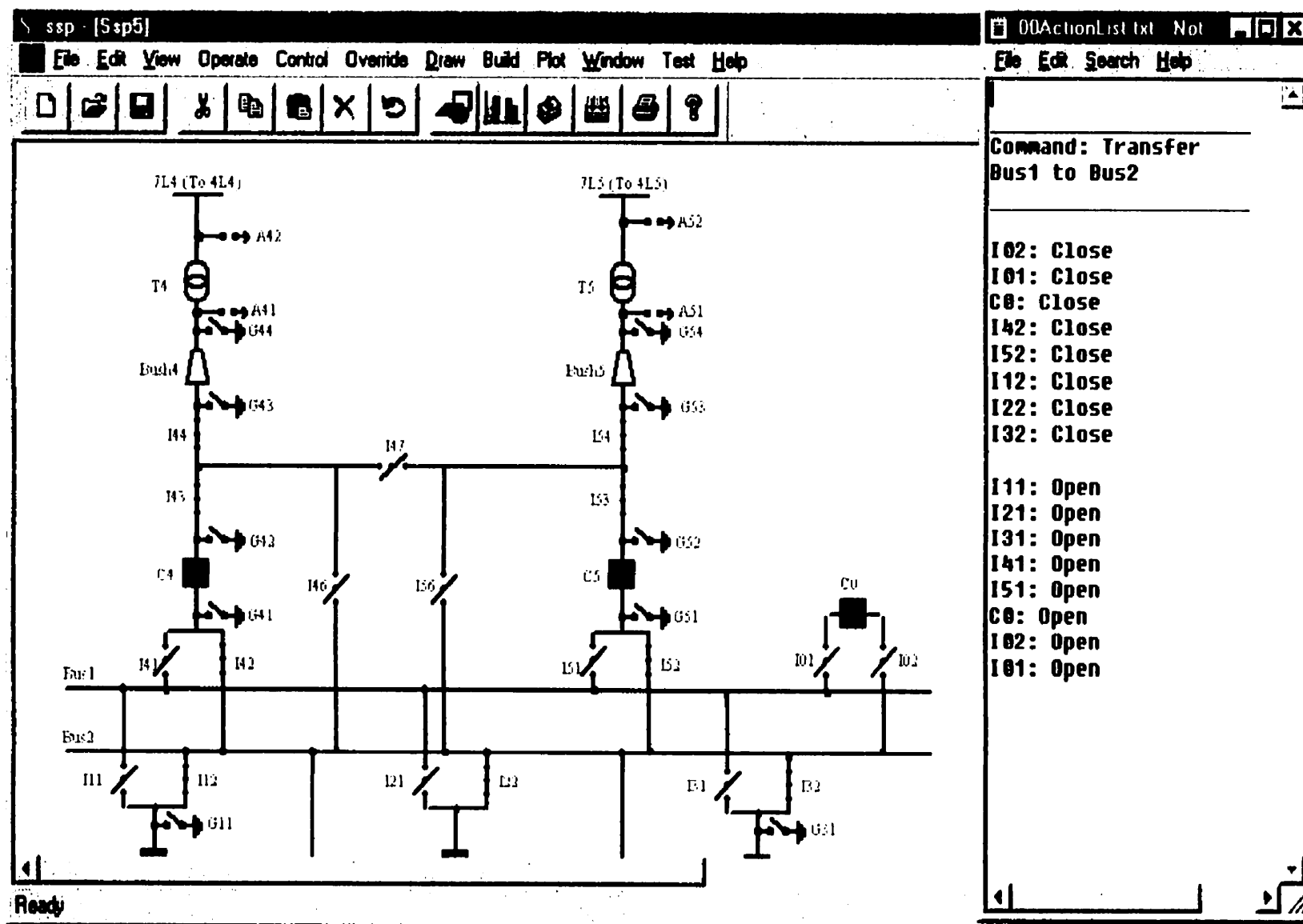


Figure 7.20. Command being issued to transfer lines from Bus1 to Bus2.



**Figure 7.21.** The substation after the SSP transfers the lines from Bus1 to Bus2.

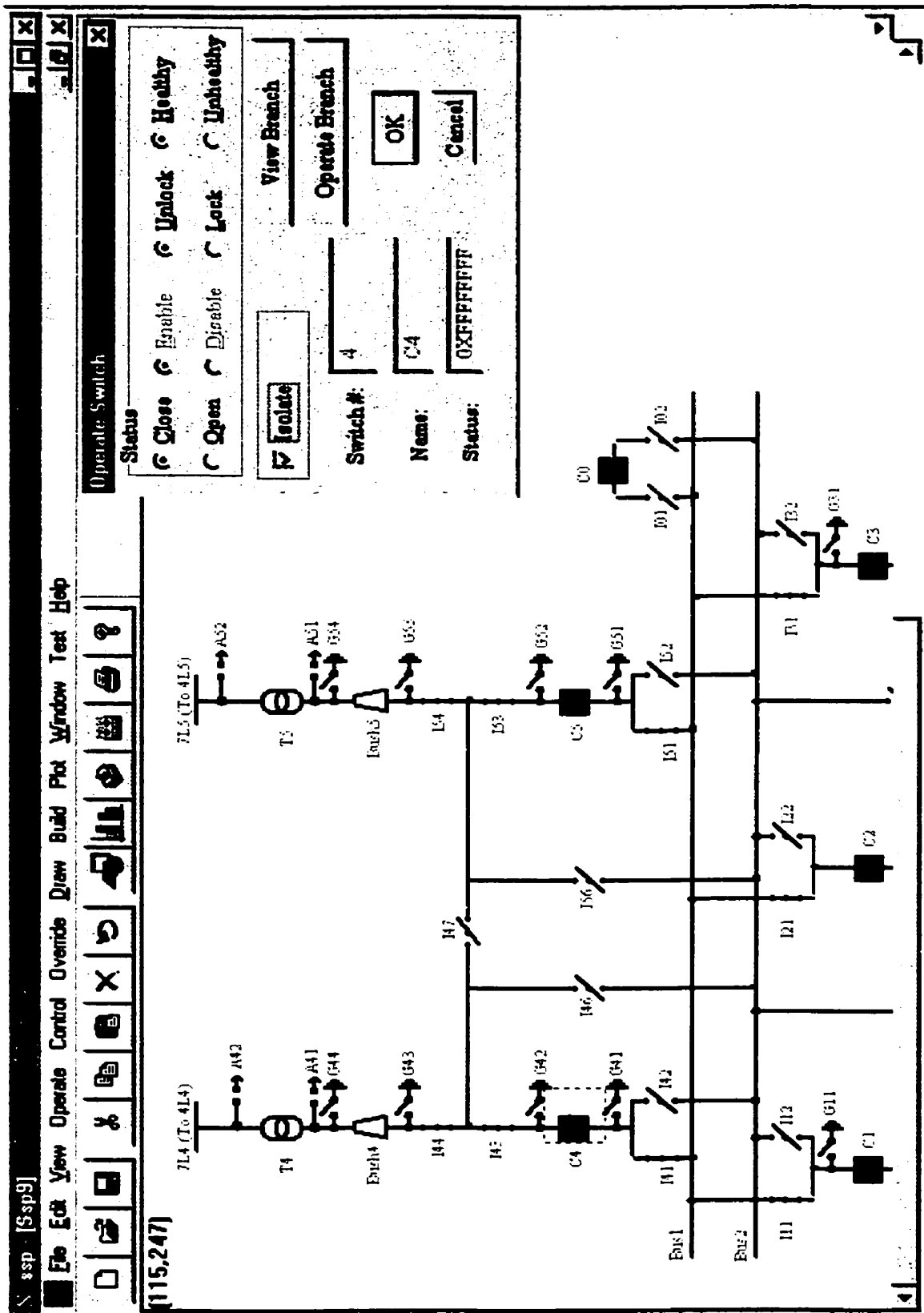
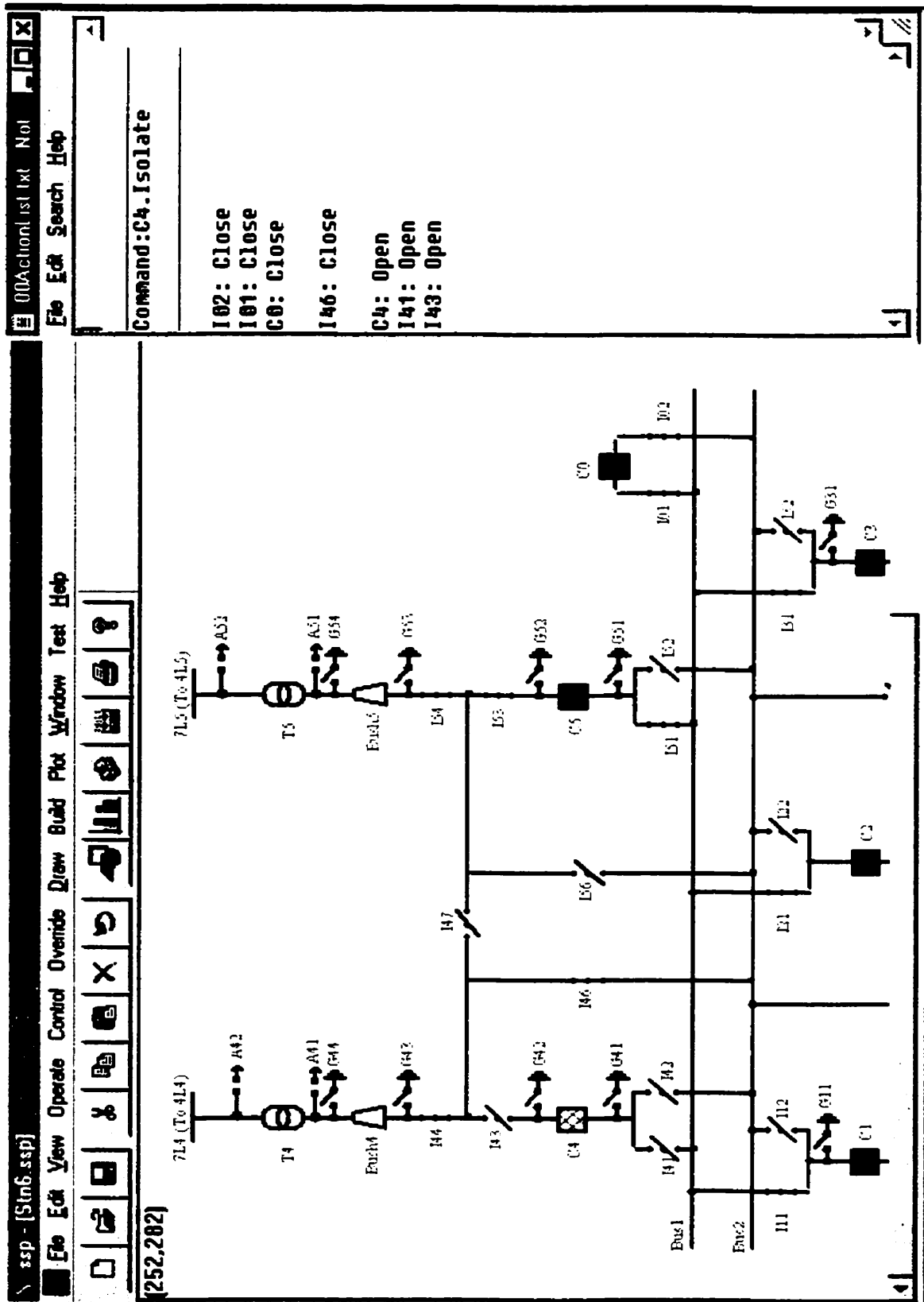


Figure 7.22. Command is being issued to isolate circuit breaker C4.



**Figure 7.23.** The substation after the SSP isolated circuit breaker C4.

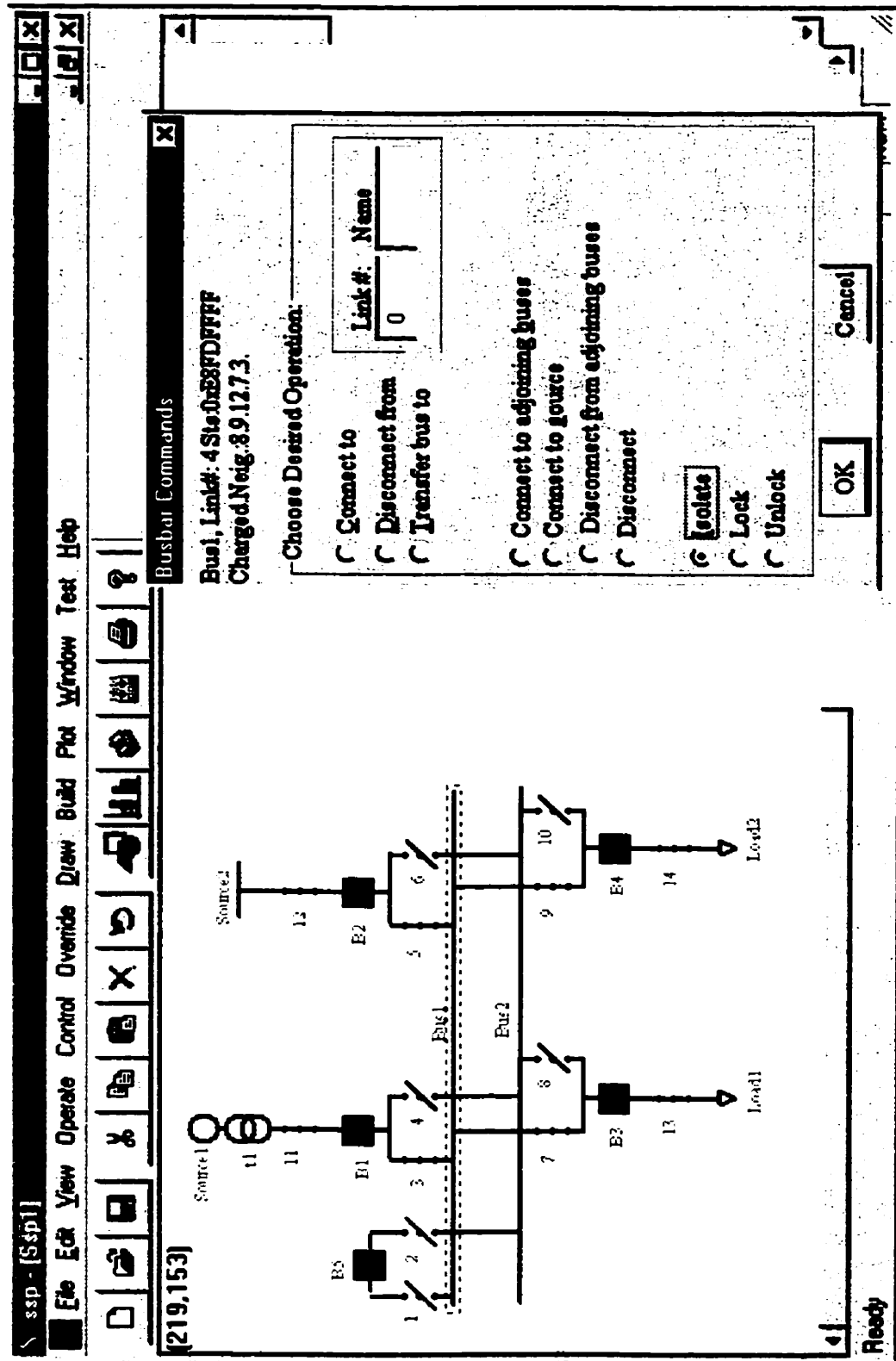


Figure 7.24. Command being issued to isolate Bus1 on Test Substation 1.

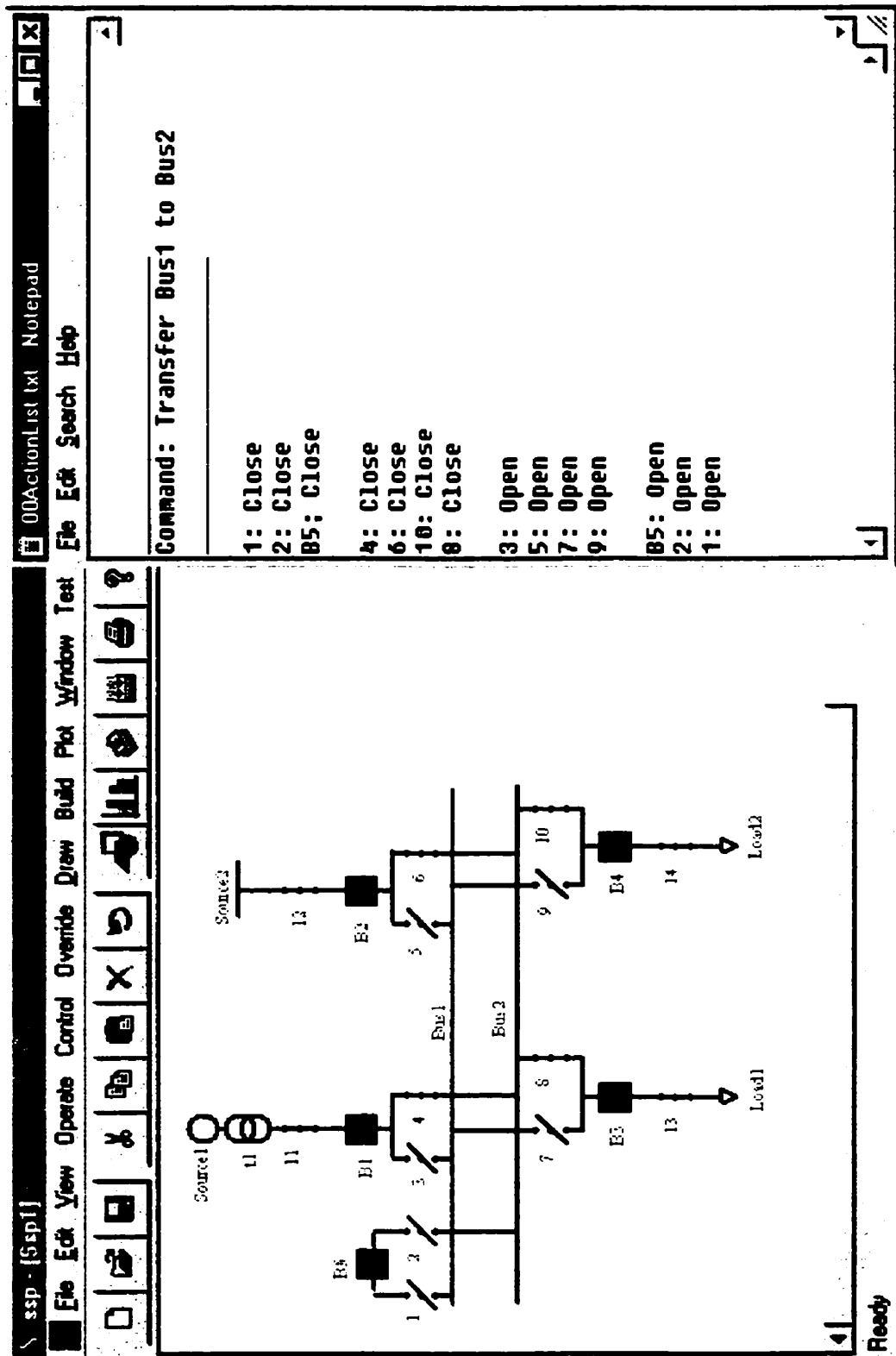


Figure 7.25. The state of the substation after isolating Bus1.

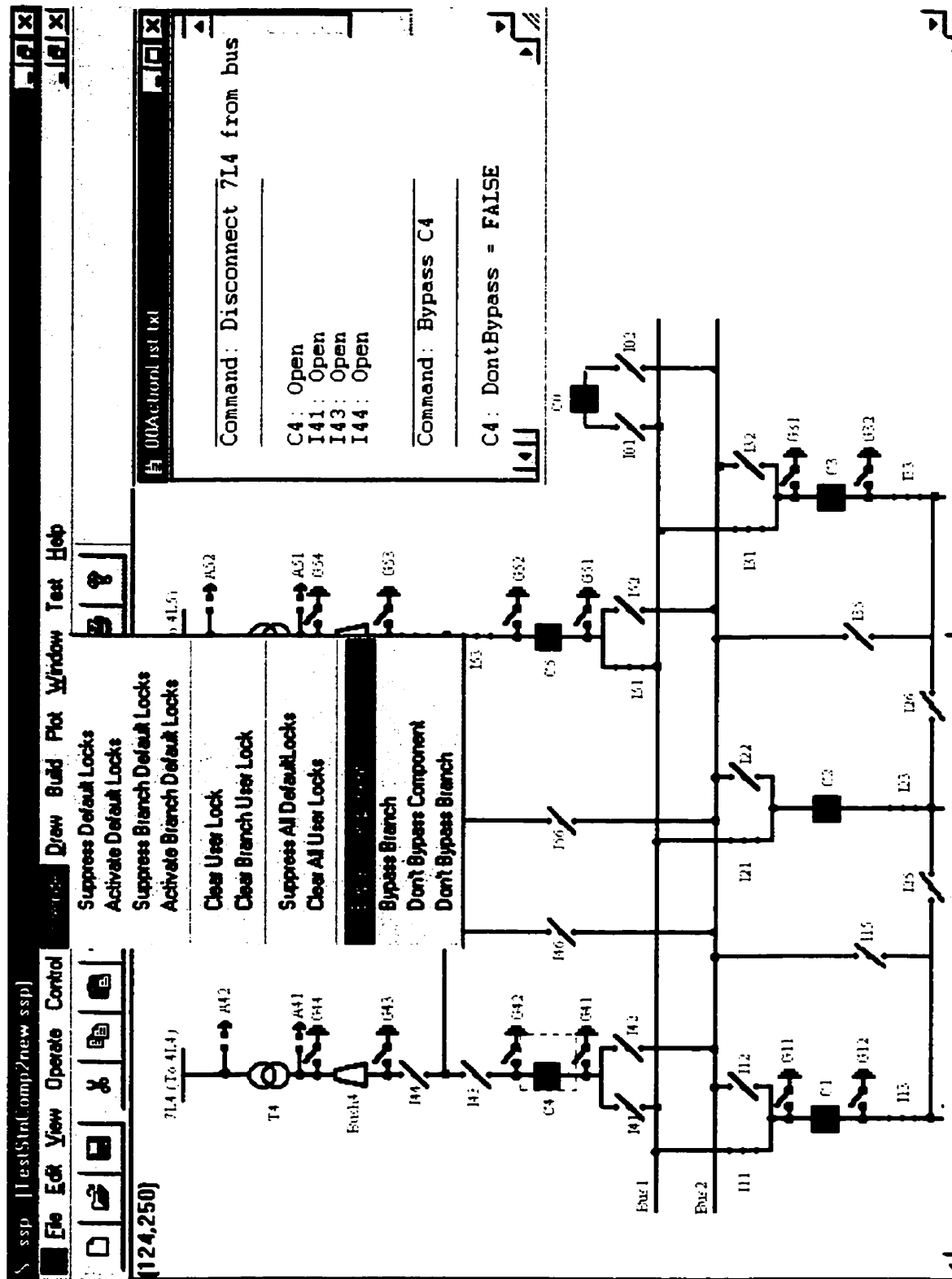


Figure 7.26. Command being issued to bypass circuit breaker C4 from power flow paths to be formed in future.



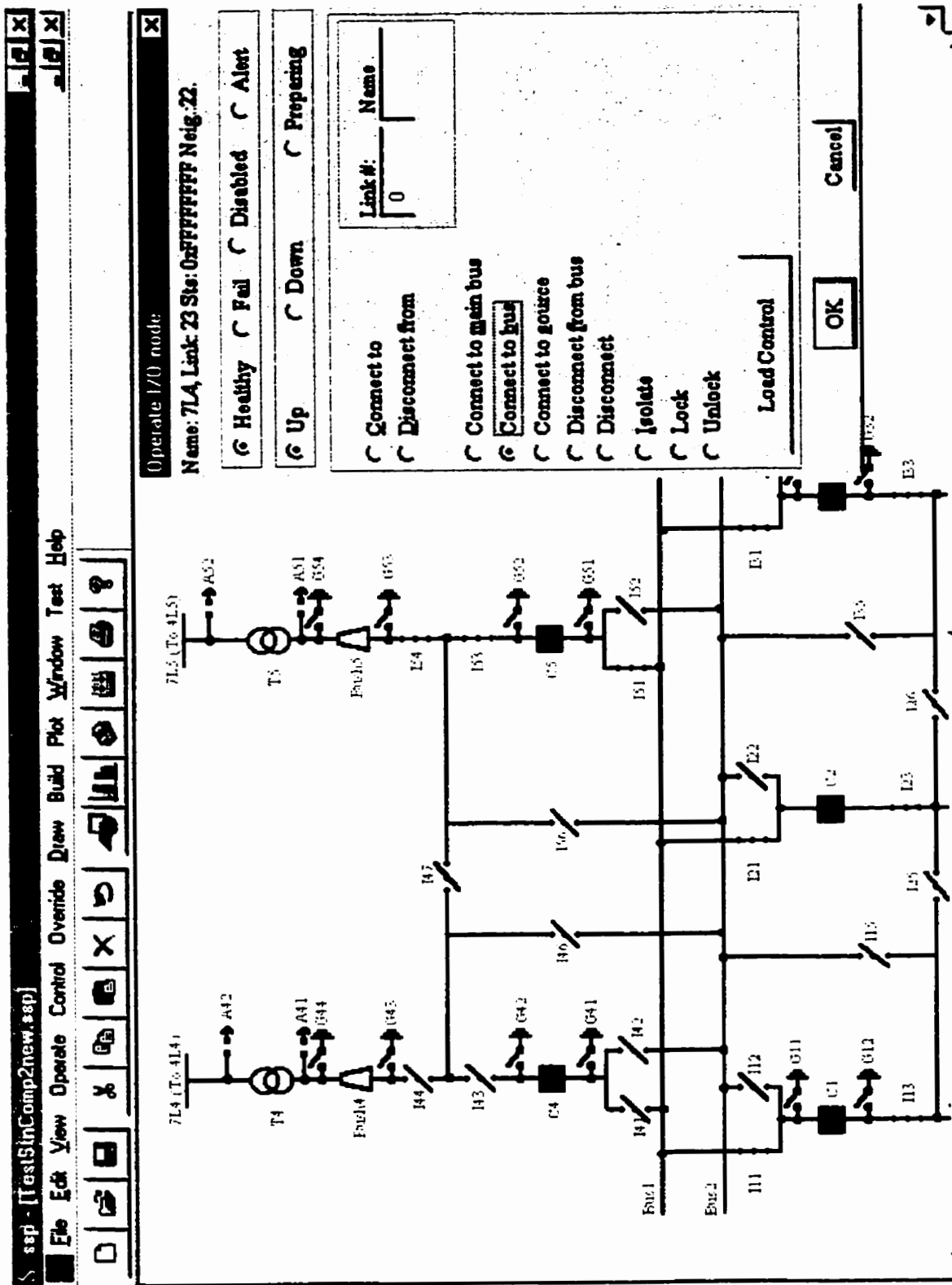


Figure 7.27. Command being issued to connect line 7L4 to the busbar after issuing the command to bypass C4.

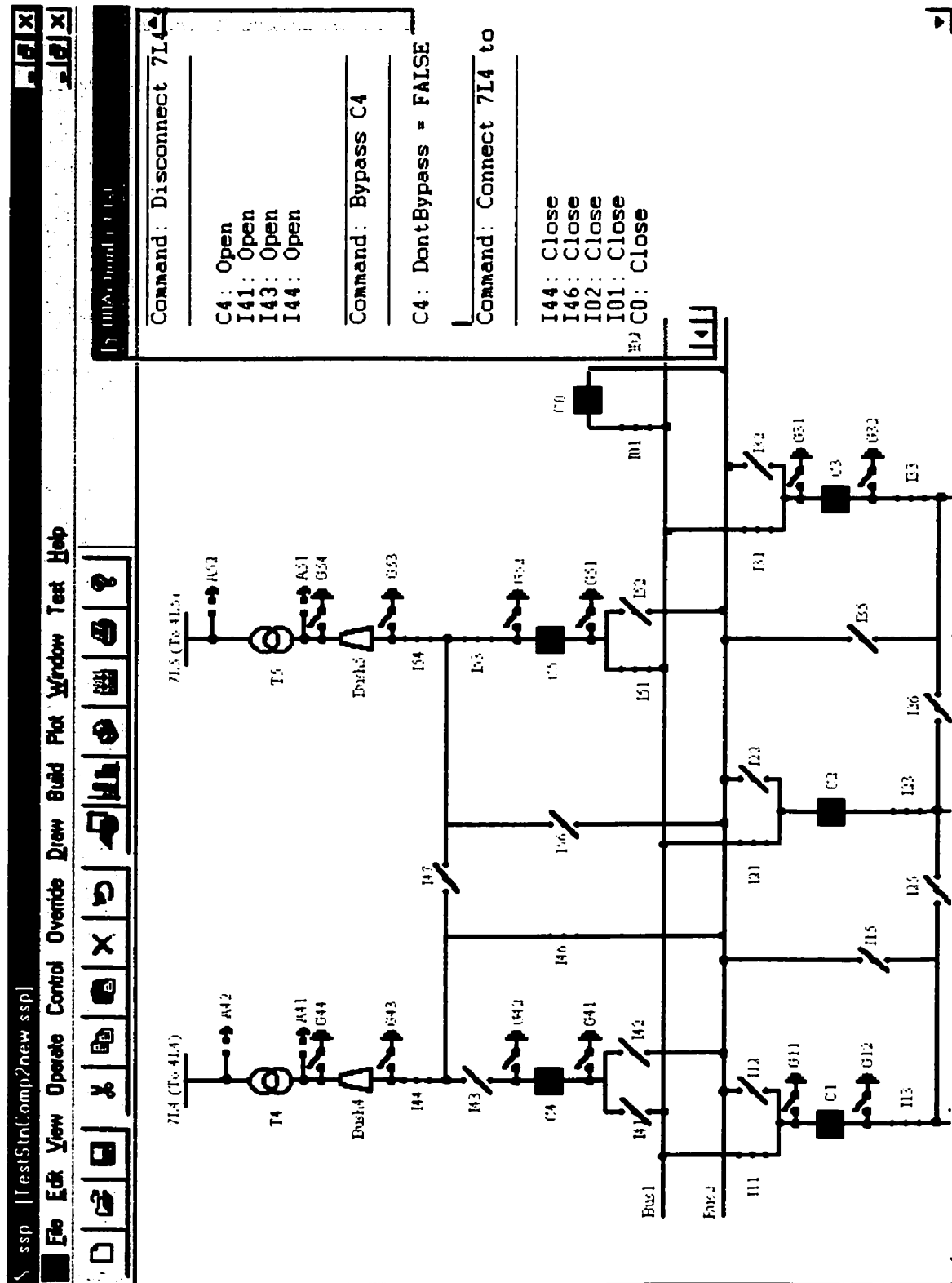


Figure 7.28. Connection made between line 7L4 and Bus1 while bypassing C4.

## **8. SUMMARY, CONCLUSIONS AND FUTURE WORK**

The research and development of computer aided design of substation switching schemes have been presented in Chapters 1 to 7. The summary of the work presented in this thesis is provided in this chapter. The conclusions drawn from the reported activity are also given. The work that could be done in the future to enhance this development is also identified.

### **8.1 Summary**

A power system consists of a network of geographically distributed generating stations and load centres that are connected by transmission lines. The power system network is normally designed to permit the transmission of power on alternative routes if a line or a power equipment is out of service. The routing is done by using switches installed in substations.

Several transmission and distribution lines come together at a substation, where they meet through strategically placed isolating switches, circuit breakers and grounding switches. The cost of a substation and the flexibility in directing power through alternative connections depend on the arrangements of switches and busbars. The study of the nature of these switching arrangements and their operations, and the development of switching schemes that help maintain the continuity of power supply has been the focus of the research work reported in this thesis. The objective of the research work has been defined in Chapter 1. Also reported in this chapter are the past and present trends in the practices

in operating substations.

The composition of substations, their role and their typical placements in a power system have been introduced in Chapter 2. Main switching components of a substation have been introduced and components of interest to this project have been described. Busbar configurations used in substations have also been described in the chapter.

Switching operations in a substation are carried out for many reasons, which include disconnecting faulty equipment from the rest of the system, preparing the substation for repair or maintenance of equipment, and bringing the system back to its normal operating state after an outage. Switches are interlocked so that they can operate only when their operation would not adversely affect the security of the system and would not expose the personnel working in that environment to any hazards. To achieve these objectives, the switches are operated in sequences that follow the interlocking rules. These issues and the summary of work in the area reported in the literature have been presented in Chapter 3.

Substations have many common characteristics that can be analysed to develop general constraints for the operation of equipment installed in them. The operating constraints of most switching components stem from their design and their placement in a substation. Chapter 4 of this thesis has presented a concise analysis of a switching scheme. The general operating constraints of the equipment have been described in the chapter. The data of circuit components, substations and power systems required for interlocking and switching purposes are described in the chapter. The steps involved in determining switching sequences, and the factors that must be considered in determining the switching sequences have also been reviewed in this chapter.

The design of a framework, which expedites the development of a switching scheme and uses the techniques described in Chapter 4, has been presented in Chapter 5. The framework consists of interface, script, interpreter, compiler and algorithms integrated such that it can automatically develop interlocking scheme and generate switching

sequences using the accepted operating practices while allowing special requirements to be included in the developed control system. A protocol for handling the status of the circuit components, substation and system properties, and a protocol for handling the interlocks and the techniques for implementing them in the development of an overall switching scheme have been described in this chapter.

The concepts described in Chapters 4 and 5 were implemented in the form of a software package called Substation Switching Scheme Development Package (SSP). Object oriented approach and C++ programming language were used in developing this package. The techniques used to develop the SSP have been described in Chapter 6. The steps involved in compiling the required data and generating switching scheme have been described in the chapter. Algorithms used for implementing various modules of SSP have been presented as well.

The computer-aided generation of substation switching schemes were successfully tested with the SSP. Switching schemes for eight different substations were developed for these tests. Several test results that demonstrate the techniques presented in this thesis have been presented in Chapter 7. The results show that the SSP is a viable tool for implementing interlocking schemes and generating switching sequences. The implementation of SSP utilizes concepts from electrical engineering, mathematics, and computer science. The theoretical foundation comes from the well known branches of these disciplines, namely power system control and protection, Boolean logic, graph theory, circuit analysis, numerical methods for solving algebraic equations and object oriented system analysis, design and implementation.

This project has shown that it is possible to develop an integrated environment for the computer-aided design, development and testing of switching schemes for substations using the concepts reported in this thesis.

## **8.2. Conclusions**

A survey of the published literature in the area of substation switching indicated that the most of the work reported in those publications address selected aspects of substation switching. None of the publications report the development of a framework that provides techniques for specifying component data, power system data, interlocking constraints and sequence switching constraints of a substation, and automatically generating a switching scheme from that information.

The work reported in this thesis identified the requirements, protocols, interfaces, script, and interpreter, and integrated them in a technique. The work studied and compiled the generally accepted interlocking and sequence-switching practices and demonstrated a technique for developing software algorithms to implement them. The work showed that it is possible to automatically take care of the generally accepted practices so that the users of the developed framework would focus mainly on the switching operations.

It is shown that a protocol for describing the status of components and algorithms for processing the topology of a substation, predicting over-voltage, under-voltage and overload conditions, determining optimal energy flow paths, and generating switching sequences can be developed. It is further shown that switching schemes for substations can be developed using those protocols and algorithms. Extensive testing showed that the protocols and algorithms work correctly.

The work reported in this thesis has demonstrated that the identified data adequately describe components, substations and power systems for the purpose of developing switching schemes. It has also been shown that the syntactic and semantic rules for describing interlocks can handle substations of any degree of complexity. The work that is reported in this thesis has shown that the procedures developed for implementing switching commands work correctly. It has also been demonstrated that it is possible to define a complex switching command in the form of a series of simple commands.

The applications reported in this thesis have demonstrated that the developed software

package correctly identifies safe switching operations from unsafe switching operations. They have also demonstrated that interlocking schemes can be developed for a substation using the approach presented in this thesis. Desired degree of flexibility or restraint can be provided while ensuring safety of the substation equipment and personnel working in those substations.

Tests showed that the SSP develops switching sequences that operate several switches in an appropriate order. They also showed that the SSP interprets the commands and determine an optimal switching sequence for implementing the command. It is also shown that the SSP has the ability to collect a log of the switching operations that were determined and implemented. The results presented in this thesis have showed that the developed technique is a viable approach that can be implemented in a software package.

### **8.3 Suggestions for Future Work**

This research work has adequately demonstrated that it is feasible to develop a computer-aided design tool for substation switching schemes. This work has developed a foundation for the development of a tool of a larger proportion. The following work could be done to enhance the work reported in this thesis.

1. Integrate the data logging mechanism with commercial databases to utilize their power in retrieving and analysing the data efficiently.
2. Add graphical means of supplying and viewing power-system data. Provide facility for visual presentation of the load-flow and system studies results along with the power system diagrams.
3. Enhance the power-system simulation modules to perform iterative sensitivity analysis until a desired alternative solution is found.
4. Develop hardware and software interface between the switching scheme and the switchyard components.
5. Incorporate real-time clock of desired accuracy and precision to manage real-time operations.

## **REFERENCES**

1. Lythall, R.T., J & P Switchgear Book, Seventh Edition, Newness-Butterworths, Borough Green, Sevenoaks, Kent, 1980.
2. Dorf, Richard D., Editor, The Electrical Engineering Handbook, CRC Press, Inc., Boca Raton, Florida, USA, 1993.
3. Dhakal, Pramod, Substation Switching, M. Sc. Thesis, College of Graduate Studies and Research, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, 1995.
4. Gönen, Turan, Electrical Power Distribution Systems, Chapter 4, McGraw Hill Book Co., Inc., 1996.
5. De Haas, J., Biolders, M.J.J., Sidler H.F.A., Vos, C.W.M and Zantinge, J.F., "Comparative Analysis of Different Types of Substation Configurations," CIGRE International Conference on Large High Voltage Electric Systems, Article No. 23-11, 1989, pp. 1-6.
6. Electrical Transmission and Distribution Reference Book, Chapter 20, Central Station Engineers of the Westinghouse Electric Corp., Pennsylvania, USA, 1964.
7. Hutchinson, V.R.D., "Interlocking in Large Electricity-supply Substations - A Fundamental Approach," Proceedings of the IEE, Vol. 113, No. 6, June 1966, pp. 1063-74.
8. Cory, B.J., "An Approach by Means of Mathematical Logic to the Switching of Power-system Networks," Proceedings of the IEE, Vol. 110, No. 1, 1963, pp. 185-96.
9. Hope, Gordon S. and Cory, Brian J., "Development of Digital Computer Programs for the Automatic Switching of Power Networks," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-87, No. 7, 1968, pp. 1587-99.



10. Row, C.H.T. and Cory, B.J., "Substation Interlocking and Switching with a Small Digital Computer," Conference on Computers in Power System Operation and Control, Bournemouth College of Technology, 1972, pp. 57-63.
11. Couch, G.H. and Morrison, I.F., "Substation Interlocking Using a Digital Computer," Paper No. 71 CP 42-PWR, IEEE Winter Power Meeting, New York, N.Y., Jan.-Feb. 1971, pp.1-8.
12. Lidgate, D., Aitchison, P.W. and Acock, D., "Electrical Interlock Design for Complex High-power Switching Networks," Proceedings of IEE, Vol. 126, No. 2, February 1979, pp. 152-58.
13. Brand, K.P., Kopiansky, J. and Wimmer, W., "Topology-based Interlocking of Electrical Substations," IEEE Transactions on Power Delivery, Vol. PWRD-1, No. 3, July 1986, pp. 118-26.
14. Couch, G.H., Morrison, I.F., "Substation Switching - An Approach to the Determination of Optimal Sequences," Fourth Power System Computation Conference (PSCC) Proceedings, Grenoble, September 1972, Paper No. 2.1/1, pp. 1-18.
15. Udo, Muneyuki, "Optimum Change-over Switching of Network in Power Systems," Fourth Power Systems Computation Conference (PSCC) Proceedings, Grenoble, September 1972, Vol.2, Paper No. 2.2/3, pp. 1-15.
16. Traca-de-Almeida, A., "Substation Switching Using a Digital Processor", Proceedings of the IEE Conference on Developments in Power System Protection, Publication No. 185, London, U.K., April 1980, pp. 59-63.
17. Traca-de-Almeida, A., "Substation Interlocking and Sequence Switching Using a Digital Computer", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 6, June 1981, pp. 3002-3007.
18. Weatherall, P.M. and Senior, G., "Microprocessor Control of Post Fault Sequence Switching," Proceedings of the IEE Conference on Developments in Power System Protection, London, Publication No. 185, April 1980, pp. 59-63.
19. Marlow D.E., Dauncey R.G., "An Approach to Sequence Switching Using Computerised Telecontrol," Proceedings of the IEE Second International Conference on Power System Monitoring and Control, Publication # 266, 1986, pp. 52-56.

20. Dialynas, E.N. and Machias, A.V., "Interactive Modelling of Substation Switching Operations Following a Failure Event," IEE Proceedings, Vol. 134, Pt. C, No. 2, March 1987, pp. 153-61.
21. Abe, T., Goto, H., Mizutori, T., and Matsuki, N., "An Expert System for Generating Switching Sequences at Substations," IEEE Proceedings of AI for Industrial Applications, 1988, pp. 326-331.
22. Zhang, Z.Z., Hope, G.S. and Malik, O.P., "A Knowledge-based Approach to Optimal Switching in Substations," IEEE Transaction on Power Delivery, Vol. 1, No. 1, Jan. 1990.
23. Schulz, Noel N.; Wollenberg, Bruce F., "Expert System to Aid Evaluation and Generation of Switching Procedures," Proceedings of the American Power Conference Proceedings of the 57th Annual American Power Conference, Part 2, Chicago, USA, Apr 18-20, 1995, pp. 1481-1486.
24. Cross, A.D., Brailsford, J.R., and Brint, A.T., "Expert Systems to Support Network Switching", Proceedings of the IEE 12<sup>th</sup> International Conference on Electricity Distribution, Publication No. 373, Birmingham, UK, May 1993, pp. 4.9.1-5.
25. Corera, J., Laresgoiti, I., Cockburn, D., and Cross, A., "A Cooperative Approach Towards the Solution of Complex Decision Problems in Energy Management and Electricity Networks", Proceedings of the IEE 12<sup>th</sup> International Conference on Electricity Distribution, Publication No. 373, Birmingham, UK, May 1993, pp. 4.19.1-6.
26. Wu, J.S., Tomsovic, K.L., and Chen, C.S., "A Heuristic Search Approach to Feeder Switching Operations for Overload, Faults, Unbalanced Flow and Maintenance," IEEE Transactions in Power Delivery, Vol. 6, NO. 4, October 1991, pp. 1579-1585.
27. Yoshizawa, J., Ogi, H., Takano, To., and Matsumoto, K., "An Automatic Knowledge Acquisition Method for Switching Sequences and its Evaluation," IEEE Transactions on Power Systems, Vol. 9, NO. 2, May 1994, pp. 884-90.
28. Dhakal, P., Sachdev, M.S., "A Computer Based Substation Switching Scheme," Proceedings of the 1995 IEEE WESCANEX Communications, Power, and Computing Conference, Winnipeg, Canada, IEEE Cat. # 95CH3581-6, Vol. 1, May 1995, pp. 30-35.
29. Sachdev, M.S., Dhakal, P., Sidhu, T.S., "A Configuration Independent

- Substation Switching Scheme," Proceedings of the 1996 Canadian Conference on Electrical and Computer Engineering, CCECE'96. Part 1, v 1, Calgary, Canada, IEEE Cat. No. 96TH8157, ISBN 0-7803-3143-5, May 1996, pp. 26-29.
30. Dy Liacco, T.E., and Kraynak, T.J., "Processing by Logic Programming of Circuit-Breaker and Protective Relaying Information," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-88, No.2, 1969, pp. 171-5.
  31. Dy Liacco, T.E., Ramarao, K.A., and Weiner, A.W., "Network Status Analysis for Real-Time Systems," PICA 8th Conference Proceedings, Minneapolis, USA, 1973, pp. 356-362.
  32. Sason, A. M., Ehrmann, S. T., Lynch, P., and Van Slyk, L. S., "Automatic Power System Network Topology Determination, IEEE Transactions on Power Apparatus and Systems," Vol. PAS-92, 1973, pp. 610-618.
  33. Couch, G. H., and Morrison, I. F., "Data Validation and Topology Determination for Power System Monitoring and Control," IEEE Power Engineering Society Summer Meeting and Energy Resources Conference, Anaheim, California, July 14-19, 1974, Paper # C 74 361-2, pp. 1-9.
  34. Sullivan, A.C., Reichert, K., Saly, S., "An On-Line Technique for Network Topology Determination for use in Security Assessment Studies," IEE International Conference on On-line Optimization of Transmission and Distribution Systems, London, England, Jan. 1976, pp. 61-66.
  35. Bertran, M. and Corbella, X., "On the Validation and Analysis of a New Method for Power Network Connectivity Determination," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, NO. 2, February 1982, pp. 316-24.
  36. Prais, Mauro, Bose, Anjan, "A Topology Processor that Tracks Network Modifications Over Time," IEEE Transactions on Power Systems, Vol. 3, No. 3, August 1988, pp. 992-998.
  37. Scott, B., "Review of Load-Flow Calculation Methods," Proceedings of the IEEE, Vol. 62, No. 7, July 1974.
  38. Booch, G., Object Oriented Analysis and Design with Applications, 2nd Edition, Benjamin Cummings, Redwood City, USA, 1994.
  39. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W., Object-Oriented Modeling and Design, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1991.

40. Meyer, B., Object Oriented Software Construction, Prentice Hall Inc., USA, 1989.
41. Stroustrup, B., The C++ programming language, 2nd ed, Addison-Wesley, Reading, USA, 1992.
42. Sachdev, M.S., Dhakal, P., and Sidhu, T.S., "Design Tool Generates Substation Interlock Schemes", IEEE Computer Applications in Power, Vol.13, No. 2, April 2000, pp. 37-42.
43. Sachdev, M.S., Dhakal, P., and Sidhu, T.S., "A Computer-Aided Technique for Generating Substation Interlocking Schemes" IEEE Transactions on Power Delivery, Vol. 15, No. 2, April 2000, pp. 538-44.
44. Sachdev, M.S., Dhakal, P., Sidhu, T.S., Automated Design of Substation Switching Systems, IEE Publication n. 434, March 1997, pp. 369-372.
45. Dhakal, P., Sachdev, M.S., Sidhu, T.S., SSP User's Manual, Power Systems Research Group, University of Saskatchewan, Saskatchewan, Canada, 1998.
46. Lee, S.J., Liu, C.C, Venkata, S.S., "Extended Expert System for Service Restoration of Distribution Feeders", IFAC Proceedings Series, No. 9, Pergamon Press Inc., August/September 1989, pp. 423-29.
47. Strunge, C., "Applying Array-based Logic to Substation Control for Switch Interlocking", IEEE Transactions on Power Delivery, Vol. 14, No. 3, 1999, pp. 879-88.
48. "Common Data Format for the Exchange of Solved Load Flow Data", Working Group on a Common Format for the Exchange of Solved Load Flow Data, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-92, No. 6, November/December 1973, pp. 1916-1925.
49. Grainger, John J. and Stevenson, William D. Jr, "Power system analysis", McGraw-Hill, New York, 1994
50. Kreyszig, Erwin, "Advanced Engineering Mathematics", John Wiley and Sons, Inc., New York, 1988.
51. Tremblay, Jean-Paul and Sorenson, P.G., "An Introduction to Data Structures With Applications", McGraw-Hill, New York, 1984.

## Appendix A

### Decoupled Power Flow Calculations

A power system represents an electric network that is composed of branches and nodes. Typically, transmission lines constitute the branches and buses constitute the nodes. Power is injected into the network at some buses and is tapped out by system loads at other buses. At any moment the generation is equal to the demand. If too much or too little load is connected to the system, the system voltage levels cannot remain at normal level. Also some transmission lines may carry more than rated power under abnormal operating conditions. Power flow calculations [37, 49] are performed to predict the line currents and bus voltages when performing switching operations in a power system.

The active and reactive powers injected into bus  $i$  of a power system is equal to the sum of the flows of power in the lines connected to that bus. This can be written as follows:

$$\begin{aligned}
 P_i + jQ_i &= \sum_{j=1}^N (P_{ij} + jQ_{ij}) \\
 &= V_i \sum_{j=1}^N I_{ij}^* \quad , i = 1, 2, \dots, N \quad (A.1)
 \end{aligned}$$

where,

- $N$  is the number of buses,
- $V_i$  is the voltage at bus  $i$ ,
- $I_{ij}$  is the current flowing from bus  $i$  to bus  $j$ ,
- $P_i$  is the active power flowing into bus  $i$ ,
- $Q_i$  is the reactive power flowing into bus  $i$ ,
- $P_{ij}$  is the active power flowing from bus  $i$  to bus  $j$ , and
- $Q_{ij}$  is the reactive power flowing from bus  $i$  to bus  $j$ .

The active and reactive powers flowing into a bus  $i$  of an  $N$  bus system can be now written by expanding Equation A.1 as follows.

$$P_i = |V_i| \sum_{j=1}^N |V_j| (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) , \quad i = 1, 2, \dots, N \quad (\text{A.2})$$

$$Q_i = |V_i| \sum_{j=1}^N |V_j| (G_{ij} \sin \delta_{ij} + B_{ij} \cos \delta_{ij}) , \quad i = 1, 2, \dots, N \quad (\text{A.3})$$

where:

$|V_i|$  is the magnitude of the voltage at bus  $i$ ,

$|V_j|$  is the magnitude of the voltage at bus  $j$ ,

$\delta_{ij}$  is the phase angle difference between the voltages at buses  $i$  and  $j$ ,

$G_{ij} + jB_{ij}$  is the element of the bus admittance matrix in row  $i$  and column  $j$ ,

One equation of the form of Equation A.2 is prepared for each bus. This provides one set of  $N$  nonlinear algebraic equations. Similarly a set of  $(N-M+1)$  equation of the form of Equation A.3 are prepared, where  $M$  is the number of generator-buses.

In these equations, the admittance of the line connecting two buses is constant. The active and reactive power flowing from one bus to another are, therefore, functions of the phase angles and magnitudes of the bus voltages.

The left hand side of Equations A.2 and A.3 are the scheduled power injections and the right hand sides of these equations are the calculated power flows in the power-flow solution. Before the solution is obtained, the scheduled power injections and the sum of the calculated power flows out of a bus are not equal. The differences are the mismatches that can be expressed as

$$\Delta P_i = P_i(\text{Scheduled}) - P_i(\text{Calculated}) \quad (\text{A.4})$$

$$\Delta Q_i = Q_i(\text{Scheduled}) - Q_i(\text{Calculated}) \quad (\text{A.5})$$

The calculated power flows depend on the magnitudes and phase angles of the voltages. The corrections that must be applied to the voltage magnitudes and phase angles can be determined by solving the following equations.

$$\Delta P_i = \sum_{j=1}^N \frac{\partial P_i}{\partial \delta_j} \Delta \delta_j + \sum_{j=1}^{N-M+1} \frac{\partial P_i}{\partial |V_j|} \Delta |V_j|, \quad i = 1, 2, \dots, N \quad (\text{A.6})$$

$$\Delta Q_i = \sum_{j=1}^N \frac{\partial Q_i}{\partial \delta_j} \Delta \delta_j + \sum_{j=1}^{N-M+1} \frac{\partial Q_i}{\partial |V_j|} \Delta |V_j|, \quad i = 1, 2, \dots, (N-M+1) \quad (\text{A.7})$$

After writing equations representing the mismatch of power between the scheduled and calculated values, like Equations A.6 and A.7, are written for all nodes the entire set of equations can be written in the following form:

$$\begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \dots \\ \Delta P_{N-1} \\ \Delta Q_1 \\ \Delta Q_2 \\ \dots \\ \Delta Q_{N-1} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & \dots & H_{1,N-1} & N_{11} & N_{12} & \dots & N_{1,N-1} \\ H_{21} & H_{22} & \dots & H_{2,N-1} & N_{21} & N_{22} & \dots & N_{2,N-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ H_{N-1,1} & H_{N-1,2} & \dots & H_{N-1,N-1} & N_{N-1,1} & N_{N-1,2} & \dots & N_{N-1,N-1} \\ J_{11} & J_{12} & \dots & J_{1,N-1} & L_{11} & L_{12} & \dots & L_{1,N-1} \\ J_{21} & J_{22} & \dots & J_{2,N-1} & L_{12} & L_{22} & \dots & L_{2,N-1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ J_{N-1,1} & J_{N-1,2} & \dots & J_{N-1,N-1} & L_{N-1,1} & L_{N-1,2} & \dots & L_{N-1,N-1} \end{bmatrix} \times \begin{bmatrix} \Delta \delta_1 \\ \Delta \delta_2 \\ \dots \\ \Delta \delta_{N-1} \\ \Delta |V_1|/|V_1| \\ \Delta |V_2|/|V_2| \\ \dots \\ \Delta |V_{N-1}|/|V_{N-1}| \end{bmatrix} \quad (\text{A.8})$$

These equations are often denoted in the following form:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V/|V| \end{bmatrix} \quad \text{.....(A.9)}$$

Where,

- H, N, J and L are the submatrices of the Jacobian matrix.
- $\Delta P$  is the vector of active power mismatches.
- $\Delta Q$  is the vector of reactive power mismatches.
- $\Delta \delta$  is the vector of phase angle corrections.
- $\Delta V$  is the vector of voltage magnitude corrections.

The elements of the H, J, N and L sub-matrices are calculated as follows:

Off diagonal elements ( $i \neq j$ ):

$$H_{ij} = \frac{\partial P_i}{\partial \delta_j} = -|V_i||V_j|(G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) \quad (\text{A.10})$$

$$N_{ij} = \frac{\partial P_i}{\partial V_j} = -|V_i||V_j|(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) \quad (\text{A.11})$$

$$J_{ij} = \frac{\partial Q_i}{\partial \delta_j} = |V_i||V_j|(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) \quad (\text{A.12})$$

$$L_{ij} = \frac{\partial Q_i}{\partial V_j} = -|V_i||V_j|(G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) \quad (\text{A.13})$$

The diagonal elements evaluate to as follows:

$$H_{ii} = \frac{\partial P_i}{\partial \delta_i} = |V_i|^2 B_{ii} + Q_i \quad (\text{A.14})$$



$$N_{ii} = \frac{\partial P_i}{\partial V_i} = -|V_i|^2 G_{ii} - P_i \quad (\text{A.15})$$

$$J_{ii} = \frac{\partial Q_i}{\partial \delta_i} = |V_i|^2 G_{ii} - P_i \quad (\text{A.16})$$

$$L_{ii} = \frac{\partial Q_i}{\partial V_i} = |V_i|^2 B_{ii} - Q_i \quad (\text{A.17})$$

This is the basis of the Newton-Raphson Power-Flow Technique. The power flow calculations start with some guess values of the voltage and phase angles. Correction factors for the approximated values are then determined using equation (A.8). The guessed values are updated with the correction factors and the power mismatch is calculated again. The correction in the approximated values is done iteratively till the power mismatch falls below a tolerable limit. Once the magnitude and phase angles of the bus voltages are determined, the line currents and power flows are calculated as the line admittances are known.

### **Decoupled Power Flow Method**

In most practical situations, the changes in phase angles of the bus voltages significantly affect the active power flowing in the lines. Similarly, the changes in bus voltage magnitudes significantly affect the reactive power flow in the system. However, the relationship between the voltage phase angle with the reactive power flow and also between the voltage magnitude and active power flow are very weak. Based on these observations the Jacobian matrices N and J in equation A.9 can be ignored while getting reasonably good approximations of the active and reactive power mismatches.

The Decoupled Power-Flow Technique is obtained by modifying Equation A.9. The modification consists of neglecting all the terms in J2 and J3 that are usually very small.

Since  $\delta_{ij}$  is usually a small angle ( $\ll 20^\circ$ ) the following can be assumed without introducing large errors in calculations:

$$\cos \delta_{ij} \approx 1$$

$$G_{ij} \sin \delta_{ij} \ll B_{ij}$$

$$Q \ll |V_i|^2 B_{ii}$$

When these approximations are introduced in the calculation of the elements of Jacobians H and L, the following equations are obtained:

$$\left. \begin{aligned} H_{ij} &= |V_i| |V_j| B_{ij} \\ H_{ii} &= |V_i|^2 B_{ii} \end{aligned} \right\} \quad (\text{A.18})$$

$$\left. \begin{aligned} L_{ij} &= |V_i| |V_j| B_{ij} \\ L_{ii} &= |V_i|^2 B_{ii} \end{aligned} \right\} \quad (\text{A.19})$$

The H and L Jacobians can, therefore, be represented as follows:

$$H = L = \begin{bmatrix} |V_1| & & & \\ & |V_2| & & \\ & & \dots & \\ & & & |V_n| \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \dots & \dots & \dots & \dots \\ B_{n1} & B_{n2} & \dots & B_{nn} \end{bmatrix} \begin{bmatrix} |V_1| & & & \\ & |V_2| & & \\ & & \dots & \\ & & & |V_n| \end{bmatrix} \quad \dots\dots\dots(\text{A.20})$$

Substituting H and L in Equation A.9, the following form can be achieved for the mismatch of active and reactive power:

$$\begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \dots \\ \Delta P_n \end{bmatrix} = \begin{bmatrix} |V_1| & & & \\ & |V_2| & & \\ & & \dots & \\ & & & |V_n| \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \dots & \dots & \dots & \dots \\ B_{n1} & B_{n2} & \dots & B_{nn} \end{bmatrix} \begin{bmatrix} |V_1| * \Delta \delta_1 \\ |V_2| * \Delta \delta_1 \\ \dots \\ |V_n| * \Delta \delta_n \end{bmatrix} \quad (\text{A.21})$$

$$\begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \dots \\ \Delta Q_n \end{bmatrix} = \begin{bmatrix} |V_1| & & & \\ & |V_2| & & \\ & & \dots & \\ & & & |V_n| \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \dots & \dots & \dots & \dots \\ B_{n1} & B_{n2} & \dots & B_{nn} \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \dots \\ \Delta V_n \end{bmatrix} \quad (\text{A.22})$$

Multiplying both sides by

$$\begin{bmatrix} 1/|V_1| & & & \\ & 1/|V_2| & & \\ & & \dots & \\ & & & 1/|V_n| \end{bmatrix}$$

the following equations are obtained:

$$\begin{bmatrix} \Delta P_1 / |V_1| \\ \Delta P_2 / |V_2| \\ \dots \\ \Delta P_n / |V_n| \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \dots & \dots & \dots & \dots \\ B_{n1} & B_{n2} & \dots & B_{nn} \end{bmatrix} \begin{bmatrix} |V_1| * \Delta \delta_1 \\ |V_2| * \Delta \delta_1 \\ \dots \\ |V_n| * \Delta \delta_n \end{bmatrix} \quad (\text{A.23})$$

$$\begin{bmatrix} \Delta Q_1 / |V_1| \\ \Delta Q_2 / |V_2| \\ \dots \\ \Delta Q_n / |V_n| \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & B_{2n} \\ \dots & \dots & \dots & \dots \\ B_{n1} & B_{n2} & \dots & B_{nn} \end{bmatrix} \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \dots \\ \Delta V_n \end{bmatrix} \quad (\text{A.24})$$

The last two equations are often represented in the following form for the convenience:

$$[\Delta P/|V|] = [B'][\Delta \delta] \quad (\text{A.25})$$

$$[\Delta Q/|V|] = [B''][\Delta V] \quad (\text{A.26})$$

where  $[B']$  and  $[B'']$  are modified admittance matrices.

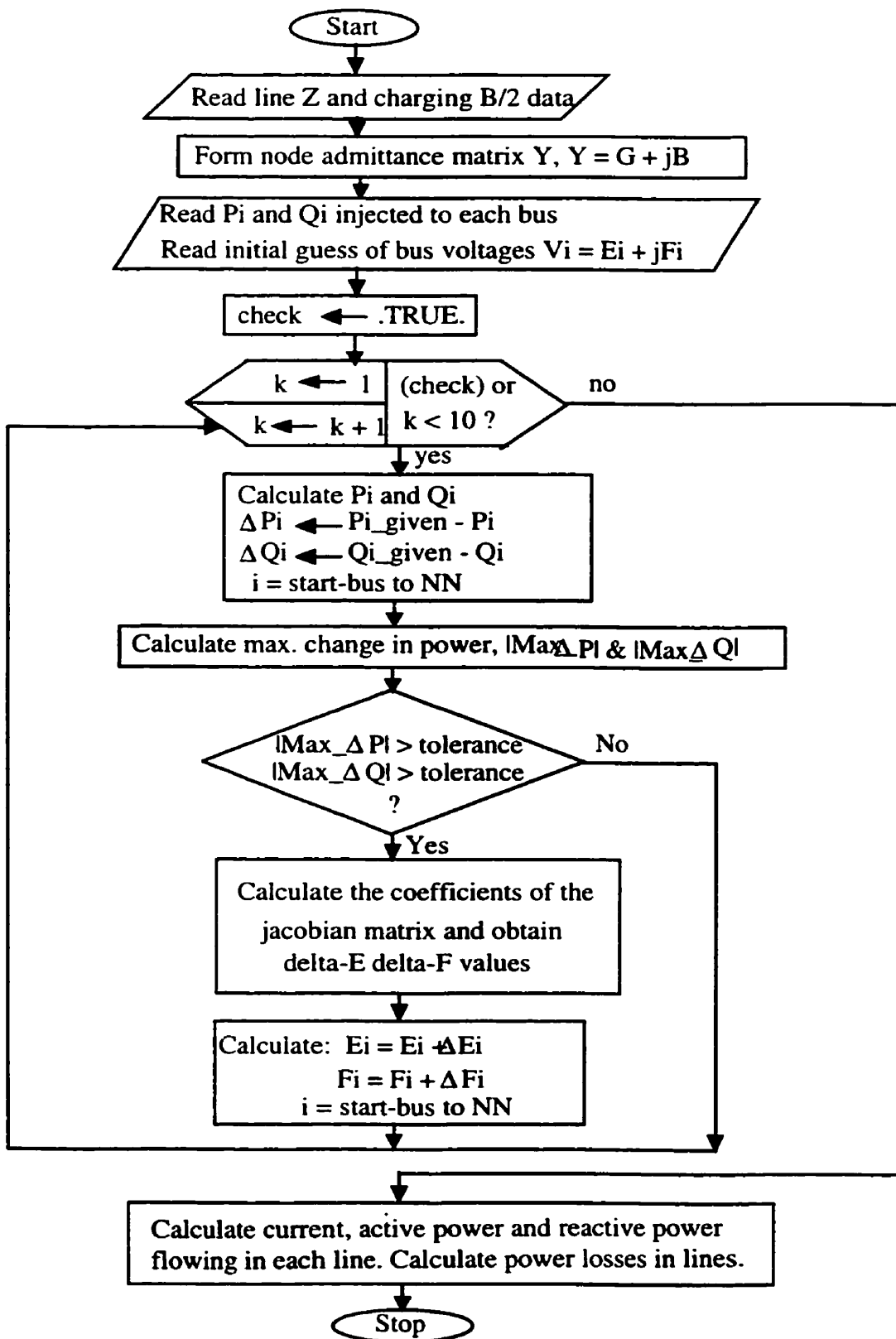
The following iterative process can be used to solve these equations.

1. Provide initial estimates of the variables  $\delta$  and  $|V|$ .

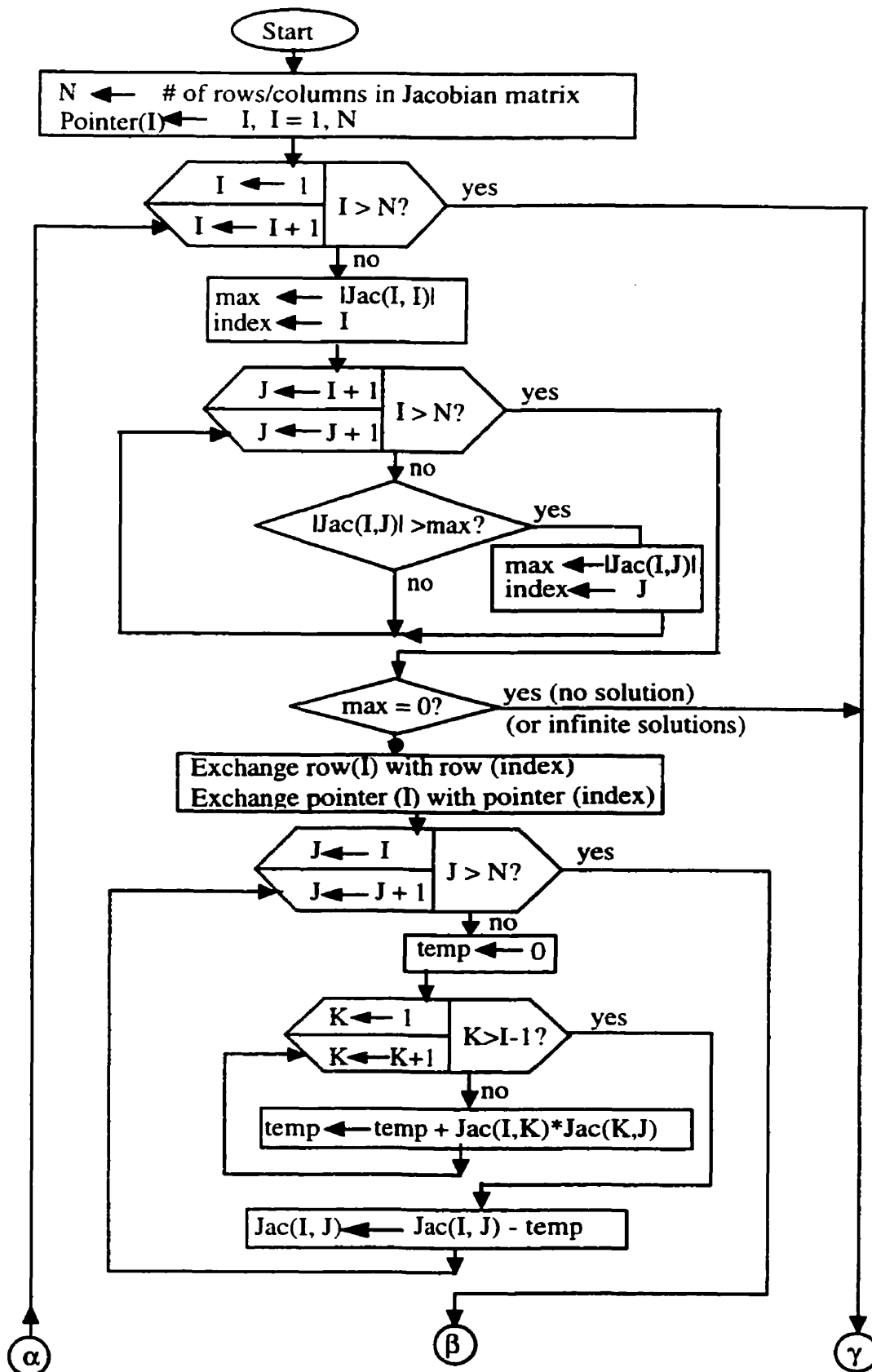
2. Using the values of  $\delta$  and  $IVI$ , calculate the active and reactive power injections using Equations A.2 and A.3.
3. Calculate  $\Delta P$  and  $\Delta Q$  using Equations A.4 and A.5. If these mismatches are within a specified tolerance, the estimates of are accepted and, therefore, the procedure is ended. Otherwise, the next step in performed.
4. Using  $\Delta P$  and  $\Delta Q$  from Step 3, calculate the corrections  $\Delta\delta$  and  $\Delta IVI$  using Equations A.23 and A.24.
5. Modify the estimated values of  $\delta$  and  $IVI$  and revert to Step 2.

Once the values of  $\delta$  and  $V$  are determined, the currents flowing in the lines are calculated. This procedure is also depicted in the flowchart shown in Figure A.1. In the flowchart,  $i$  and  $k$  variables used for traversing through the matrices.

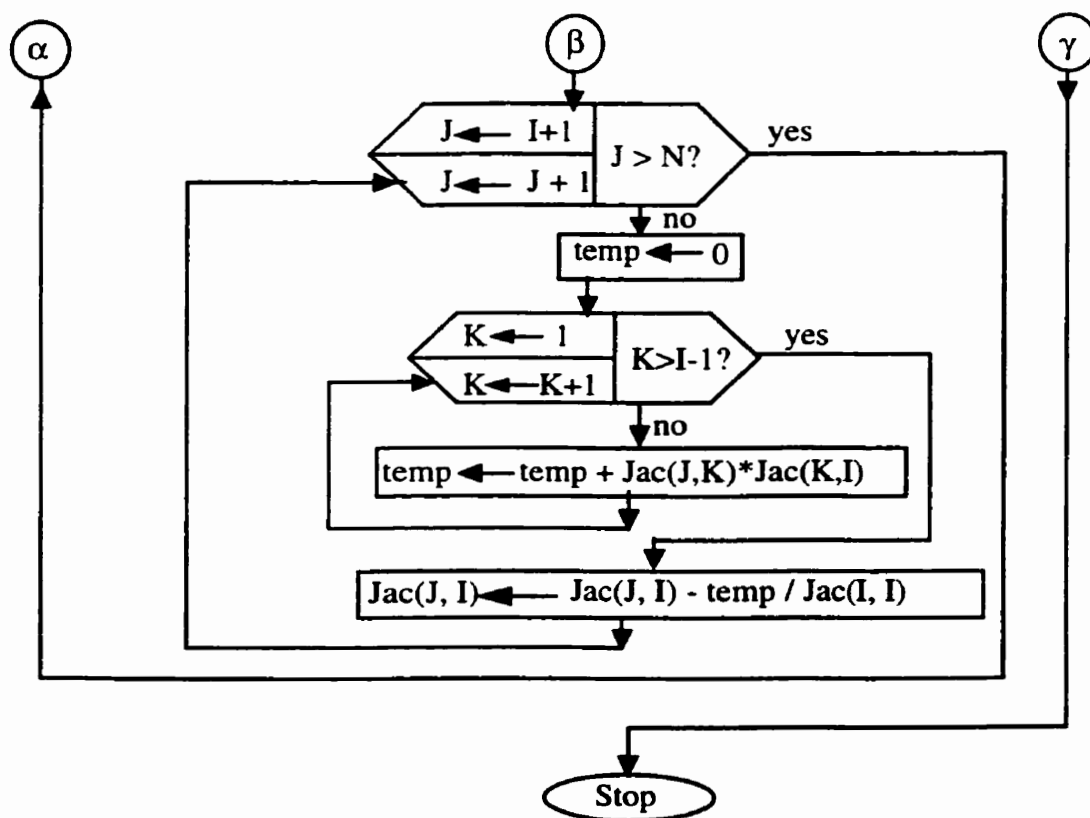
A number of approaches [50] can be used to solve the matrix equations of A.23 and A.24 to determine the estimates of  $\delta$  and  $IVI$  as mentioned in Step 5. A technique called LU factorization was used in the SSP for solving those equations. The algorithm for finding the LU factors of a matrix are is shown in Figure A.2. In the figure,  $I$ ,  $J$  and  $K$  represent temporary variables used for indexing.  $N$  is the number of rows in the Jacobian matrix (Jac) that is to be factored. Other variables have usual meanings as discussed earlier in this section.



**Figure A.1:** Procedure for load flow calculations.



**Figure A.2:** Procedure for LU factorization of Jacobian matrix (continued...).



**Figure A.2:** Procedure for LU factorization of Jacobian matrix.

## **Appendix B**

### **Selection of Programming Language and Development Environment**

C++ language was selected for the development of SSP for the following reasons:

1. C++ provides necessary language support to develop a purely object-oriented software program.
2. It is widely accepted as one of best language for serious software development.
3. It is a compiled programming language and C++ compilers are efficient and can be installed on personal computers without difficulty, unlike some other object-oriented languages, such as Smalltalk.
4. C++ compilers generate one of the most efficient executable code.
5. C++ programs can be linked easily to programs written in other languages.
6. It supports a large number of low-level functions that deal with the operating system and the hardware. This is an extremely useful feature for developing the interface between the software program residing in a computer and equipment and devices located in the switch-yard of a substation.

The SSP was written in C++ language by using Visual C++ compiler and Microsoft Foundation Classes (MFC) Library on a hardware platform provided by IBM compatible personal computers running Windows 95 operating system. The operating system and hardware were chosen for they were inexpensive compared to other systems of similar capacity. Visual C++ was chosen because of its MFC library that provides a foundation for developing Windows interface and many other container classes. MFC classes provide efficient and type-safe iterating routines. Classes from the MFC can be used to derive new classes for graphical user interfaces components that have Windows interface. Only portion of user interface of the SSP that was not developed in Visual C++ was the graph plotting program, which was developed in Visual Basic.



## **Appendix C**

### **Major Classes in the SSP**

The following classes of items are supported by the SSP.

Number	Item-Type	Class-Name	Object Notation
i.	Substation	Stn	oStn
ii.	Component	Cm	oCm
iii.	Non-component	Nc	oNc
iv.	Power System	Ps	oPs
v.	Node	Nd	oNd
vi.	Series	Se	oSe
vii.	Switch	Sw	oSw
viii.	Branch	Br	oBr

These series items were further divided into two classes, switch and non-switch. The switches were divided into the following subclasses.

Number	Item-Type	Class-Name	Object Notation
ix.	Circuit breaker	Cb	oCb
x.	Isolator	Is	oIs
xi.	Grounding Switch	Gs	oGs

The non-switch type series items were divided into the following classes.

xii.	Transformer	Tr	oTr
xiii.	Reactor	Re	oRe
xiv.	Bushing	Bh	oBh

The node type items were further specialized to form the following classes.

xv.	Busbar	Bb	oBb
xvi.	Generator	Gn	oGn
xvii.	Load	Ld	oLd
xviii.	Adjoining Substation	As	oAs

This classification breaks down the component class into a number of sub-classes as shown in Figure 6.2. Each component has other units like status management unit, an interlock unit and a command handler unit as follows:

Number	Item-Type	Class-Name	Object Notation
xix.	Status management unit	Smu	oSmu
xx.	Interlock Management Unit	Imu	oImu
xxi.	Command Handler Unit	Chu	oChu
xxii.	Windows Interface Unit	Wiu	oWiu
xxiii.	Lock Unit	Lu	oLu

The classes identified in this section form the most important of all classes defined in the application. The application uses more than a hundred classes of software elements.

## **Appendix D**

### **Graphical User Interface**

The development of a graphical user interface (GUI) is mostly system dependent due to the limitations in the available tools and techniques. The evolution of Java language has been good news in the development of the operating-system independent software tools. The Java based development tools are still maturing, and the development of SSP had no other choice than depending on system dependent tools. Competing visual graphical tools were available from vendors for different platforms that accelerate the development of GUIs. Visual C++ and Microsoft Foundation Classes (MFC) library were chosen for the development of GUI for SSP. The MFC library provided a wide variety of application interface classes, which were used as the basis for the development of the GUI.

MFC provided a wide range of basic Windows interface development utilities. It gives raw implementation of various components such as main frame window and its child windows, control bars, dialog boxes, list views, tree views, scroll bars, button controls, edit controls, hotkey controls, list controls, bitmap drawings, brushes, fonts, colour palettes, pens and brushes from the library. They were modified and assembled and linked to the application to develop highly interactive and attractive interface that conferred with the Windows operating system. This eliminated the need to go into the low level details of the operating system.

The logical, mathematical or analytical part of the switching system development tool was developed separately from its Windows interface. The visual appearance or aesthetics of the Windows interface can, therefore, be modified or enhanced as desired without interfering with the internal engine of the switching scheme development tool. In the following discussion, the part of the software tool that manages the Windows interface for the switching scheme will be referred to as Windows Main Interface Unit ( WMIU).

The WMIU works as the intermediary between the internal switching system and the user. It renders a graphical image of the substation on the screen, and interprets the user input of a component of the switching system. A WMIU contains a drawing and editing module, which allows the user to draw single-line diagrams of substations and to enter the component data. The drawing module contains a toolbox, which has symbols of various substation components. These symbols can be selected, dragged and dropped in the computer screen. A single line diagram of a substation diagram is drawn by assembling these symbols. The main switching system and the WMIU interface in such a way that when a component symbol is added or removed on the computer screen a corresponding object becomes created or deleted from the switching system.

To the eye of a user, it appears as if he/she is communicating directly with the components. A component is selected by clicking the left mouse button over it. Appropriate dialog boxes to edit data, view data or to issue commands are activated by selecting the component and then using mouse-clicks or selecting menu items. Although a user interface is an important aspect of an application like SSP, environment specific details of the GUI are avoided in this thesis. The discussion, therefore, is limited only to what types of interfaces must be provided between the internal application program and the Windows.

#### **D.1. Components of a GUI**

The MFC provides a semi-automated environment to develop the WMIU and to link menu items, tool bar items, keyboard entries and component symbols to the WIU of the concerned component. The procedures to create menus, toolbars, buttons, graphical symbols, and to link them to a particular class are described in the User's Manual of the Visual C++ and MFC. After preparing the appropriate Windows based graphical elements and linking them to appropriate WIU of the components, the user can directly interact with any desired component.

Each WIU handles five types of windows based activities. They are as follows:

1. **Symbolic Display:** Each physical component of a substation is depicted on the screen as a part of the substation diagram. The program module to draw the visual symbol of the component on the screen resides in the WIU.
2. **Data Display:** The component data are displayed on the screen by using the data display windows. These windows are dialog boxes specifically designed for each class of object, which allow the data to be displayed on the appropriate format. The display of error messages also fall in this category.
3. **Data Edit:** Each WIU has its associated dialog box which displays the component data and lets the user edit those data. The dialog boxes are used mainly for editing the properties of the components. Simple mouse click or menu based commands are implemented if the desired change in data is simply the location of the component symbol on the screen or its orientation. The graphical symbols of the components can be dragged, dropped, rotated, or flipped by using the mouse alone.
4. **Command Interface:** Each WIU has its associated dialog box which displays the list of all possible operating commands that can be issued to that component. The user can then select the desired command from the list. The WIU then identifies the command and sends it to the CHU to handle the command.
5. **Status Update:** Each WIU has an associated dialog box that allows the user to issue commands to update the status and to change selected status flags.

All major power components such as substations, switches, nodes, busbars, generators, loads, transformers and reactors have support for all five types of activities. However, only selected features are available for non-component objects such as a topology detector, text elements, simple lines and circles.

## **APPENDIX E**

### **Converting Infix Expressions into Postfix Expressions**

The conversion of an infix expression into postfix notation is implemented by using data stacks, where a directly accessible element is at its *top* and the least accessible element is at its *bottom*. The elements are pushed (inserted) or popped (deleted) from the top of the stack only.

The text based interlock expressions use LUs in them and not the normal mathematical symbols. When the IMU scans the expressions, it treats a LU as a input symbol. For example, the expression (Sw1(ON ENABLED) & Sw2(OFF)) is treated as (A & B), where A represents Sw1(ON ENABLED) and B represents Sw2(OFF). The postfix form of (A&B) becomes AB&.

The following algorithm [51], was used to convert the infix expressions into postfix notation.

1. Identify all operators and variable separators that would be used to write the expressions.
2. Determine the order of preference for processing them and then define input-precedence and stack-precedence and rank of the symbols as shown in Table E.1.
3. Initialize the stack and place a left parenthesis on it.
4. Initialize the postfix expression and the rank-count.
5. Obtain the first entry of the expression, where an entry is one of the operators, parenthesis or a LU.
6. Repeat thru Step 10 while there is still another entry.
7. Remove and output (add to postfix expression) all stack symbols whose stack-precedence values are greater than the input-precedence of the current

input symbol, obtained from Table F.1. If the rank is less than 1, then the expression is invalid and, therefore, report the error and stop processing.

8. If the current symbol and top stack symbol are matching parentheses, then pop the stack; otherwise put the current symbol on the stack.
9. Obtain the next input symbol.
10. If the stack is not empty or the rank is not equal to 1 then write invalid, otherwise write valid.

<b>Table F.1: Input-precedence, stack-precedence and rank values</b>			
<b>Symbol</b>	<b>Input-Precedence Value</b>	<b>Stack-Precedence Value</b>	<b>Rank</b>
<b>+, -</b>	<b>1</b>	<b>2</b>	<b>-1</b>
<b>*, /</b>	<b>3</b>	<b>4</b>	<b>-1</b>
<b>&amp;, !</b>	<b>5</b>	<b>6</b>	<b>-1</b>
<b>!</b>	<b>7</b>	<b>8</b>	<b>0</b>
<b>LU</b>	<b>9</b>	<b>10</b>	<b>1</b>
<b>(</b>	<b>11</b>	<b>0</b>	<b>-</b>
<b>)</b>	<b>0</b>	<b>-</b>	<b>-</b>

## **Appendix F**

### **Load Flow Data Formats**

SSP recognizes two data-formats for supplying the load-flow data. Both are based on ASCII text and one of the two must be used to supply the load flow data. One of the formats is the "IEEE Common Data Format for the Exchange of Solved Load Flow Data," which is also called CDF format. This format provides a very detailed information of the system, but requires larger data files and contains many data items whose scope falls beyond the scope of SSP. The advantage however, is that it has been known to, or used by, the utilities to exchange the load-flow data. The second format is not based on any previously established standard data format but a format developed for SSP, and it will be referred to as "Reduced Data Format," or RDF, as it contains only a subset of data required by CDF format. RDF provides just-enough information of the system required by the SSP.

This section describes these data formats. Because RDF asks only for those data that are essential for SSP to perform its load flow, this format is explained first. The following notations are used in parentheses to denote the type of data.

- |    |   |   |                                                        |
|----|---|---|--------------------------------------------------------|
| 1. | A | - | Alphanumeric (no special characters)                   |
| 2. | I | - | Integer                                                |
| 3. | F | - | Floating point                                         |
| 4. | * | - | Mandatory item                                         |
| 5. | O | - | Optional                                               |
| 6. | " | - | The text string inside the quotes be typed as they are |

#### **F.1. The Reduced Data Format**

The load flow data is supplied in a data file. The filename extensions ".RDF" and ".DAT" are the only permissible filename extensions recognized by default by SSP for RDF data files. Other file extension will be accepted only when the user asks it to accept through a File Open dialog box. The data files must be saved as ASCII text. No other data formats are recognized. The use of Tab key is not recommended.

The data are read line by line and each line must follow a definite format on how it contains its data. The lines are grouped into sections with section headers. Data items must be entered in specific columns. Blank items are permissible and are interpreted as 0, but it is advised that a 0 must be entered in the file for the sake of clarity. There can be any number of characters in a line, limited only by the editor on use. Therefore, write comments freely as you please, to clarify the data, after the last required column of the line.

### Title Data\*

=====

First line in the file.

1. Columns 0-10
2. Columns 11-end

MVA Base (F\*)

Information about the system for which the data is prepared. It is recommended that the following information be entered:

Date in format DD/MM/YY, Originator's name (A), System name (A), Other comments (A).

### Bus Data \*

=====

Section title \*:

1. Columns 1-16
2. Columns 17-end

"BUS DATA FOLLOWS" (\*)

Comment (AO)

Data count and section information\*

1. Columns 1-10
2. Columns 11-15
3. Columns 16-end

Number of buses (I\*)

"ITEMS" (O)

"Data titles: Bus Number, Bus Type, Pload, Qload, Pgenr, Voltage, Vmin, Vmax, Pgmin, Pgmax" (O)

Bus data\*

1. Columns 1- 4
2. Columns 5-8

Bus number (I\*)

Bus type (I\*, permissible values 0 to 3)

The bus must be of one of the following four types:

0 - Unregulated (load, PQ bus)

1 - Hold MVAR within voltage limits, (PQ bus)

2 - Hold voltage within VAR limits (Generator, PV)

3 - Hold voltage and angle (Swing/slack bus bus, V-Theta) (must always have one)

3. Columns 9-14

Active power, P, of the load in per unit (F\*)

4. Columns 17-23

Reactive power, Q, of the load in per unit (F\*)

5. Columns 25-29

Active power, P, of the generator in per unit (F\*)

6. Columns 33-37

Voltage in per unit (F\*)

7. Columns 41-46

Minimum permissible voltage in per unit (F)

8. Columns 49-53

Maximum permissible voltage in per unit (F\*)

9. Columns 55-61

Minimum permissible active power generation in per unit (F\*, not applicable for PQ buses)

10. Columns 63-70

Maximum permissible active power generation in per unit (F\*, not applicable for PQ buses)

11. Columns 72-77

Base kV (FO), without this, values of other quantities, such as power, will be known in per unit.

12. Columns 80-end

Comments (AO)

Section end marker\*

1. Columns 1- 4

"-999" (\*)

### Branch Data \*

#### Section title\*:

1. Columns 1-16 "BRANCH DATA FOLLOWS" (\*)
2. Columns 17=end Comments (AO)

#### Data count and section information\*

1. Columns 1-10 Number of branches (I\*)
2. Columns 11-15 "ITEMS" (O)
3. Columns 16=end "Data titles: Line number, Tap Side Bus, Load Side Bus, Type, R, X, B/2, Tap, Rating" (O)

#### Branch data\*

1. Columns 1- 4 Line number (I\*)
2. Column 5- 8 Tap side bus number (I\*); the tap side of the model for transformers or phase shifters at which the non-unity tap is on.
3. Columns 9-12 Z bus number (I\*); The side of the model for transformers and phase shifters at which the the device impedance is on.
4. Columns 13-16 Type (I\*)  
0 - Transmission line  
1 - Fixed tap  
2 - Variable tap for voltage control  
3 - Variable tap (turns ratio) for MVAR control  
4 - Variable phase angle for MW control (phase shifter)
5. Columns 17-22 Resistance, R, of the line in per unit (F\*)
6. Columns 25-30 Series reactance, X, of the line in per unit (F\*)
7. Columns 33-38 Line charging susceptance B/2 (F\*)
8. Columns 41-45 Tap setting (transformer turns ratio) (F\*)
9. Columns 47-54 MVA rating expressed in per unit (F\*)

#### Section end marker\*

- 1 Columns 1- 4 "-999" (\*)

## F.2. The IEEE Common Data Format

Only a partial description of the IEEE Common Data Format for the exchange of solved load flow data is provide in this manual. The complete description can be found in the reference [46]. The data file has lines of up to 128 characters. The lines are grouped into sections with section headers. Data items are entered in specific columns. No blank items are allowed, enter zeros instead. Floating point items should have explicit decimal point.



## Title Data

=====

First line in file.

- |    |               |                                                                                                  |
|----|---------------|--------------------------------------------------------------------------------------------------|
| 1. | Columns 2- 9  | Date, in format DD/MM/YY with leading zeros. If no date provided, use 0b/0b/0b where b is blank. |
| 2. | Columns 11-30 | Originator's name (A)                                                                            |
| 3. | Columns 32-37 | MVA Base (F*)                                                                                    |
| 4. | Columns 39-42 | Year (I)                                                                                         |
| 5. | Column 44     | Season (S - Summer, W - Winter)                                                                  |
| 6. | Column 46-73  | Case identification (A)                                                                          |

## Bus Data \*

=====

Section start line \*:

- |    |               |                      |
|----|---------------|----------------------|
| 1. | Columns 1-16  | "BUS DATA FOLLOWS" * |
| 2. | Columns 36-50 | Number of items (I*) |
| 3. | Columns 51-55 | ITEMS                |

Bus data lines \*:

- |     |                 |                                                                                                 |
|-----|-----------------|-------------------------------------------------------------------------------------------------|
| 1.  | Columns 1- 4    | Bus number (I) *                                                                                |
| 2.  | Columns 7-17    | Name (A) (left justify) *                                                                       |
| 3.  | Columns 19-20   | Load flow area number (I) Don't use zero! *                                                     |
| 4.  | Columns 21-23   | Loss zone number (I)                                                                            |
| 5.  | Columns 25-26   | Type (I) *                                                                                      |
|     |                 | 0 - Unregulated (load, PQ)                                                                      |
|     |                 | 1 - Hold MVAR generation within limits, (PQ)                                                    |
|     |                 | 2 - Hold voltage within VAR limits (gen, PV)                                                    |
|     |                 | 3 - Hold voltage and angle (swing, V-Theta) (must always have one)                              |
| 6.  | Columns 28-33   | Final voltage, p.u. (F) *                                                                       |
| 7.  | Columns 34-40   | Final angle, degrees (F) *                                                                      |
| 8.  | Columns 41-49   | Load MW (F) *                                                                                   |
| 9.  | Columns 50-59   | Load MVAR (F) *                                                                                 |
| 10. | Columns 60-67   | Generation MW (F) *                                                                             |
| 11. | Columns 68-75   | Generation MVAR (F) *                                                                           |
| 12. | Columns 77-83   | Base KV (F)                                                                                     |
| 13. | Columns 85-90   | Desired volts (pu) (F) (This is desired remote voltage if this bus is controlling another bus.) |
| 14. | Columns 91-98   | Maximum MVAR or voltage limit (F)                                                               |
| 15. | Columns 99-106  | Minimum MVAR or voltage limit (F)                                                               |
| 16. | Columns 107-114 | Shunt conductance G (per unit) (F) *                                                            |
| 17. | Columns 115-122 | Shunt susceptance B (per unit) (F) *                                                            |
| 18. | Columns 124-127 | Remote controlled bus number                                                                    |

Section end line:

- |    |              |         |
|----|--------------|---------|
| 1. | Columns 1- 4 | "-999"* |
|----|--------------|---------|

**Branch Data \***

=====

**Section start line \*:**

1. Columns 1-16
2. Columns 36-50
3. Column 51-55

**BRANCH DATA FOLLOWS\***

Number of items (I)  
"ITEMS"

**Branch data lines \*:**

1. Columns 1- 4
2. Columns 6- 9
3. Columns 11-12
4. Columns 13-14
5. Column 17
6. Column 19
7. Columns 20-29
8. Columns 30-40
9. Columns 41-50
10. Columns 51-55
11. Columns 57-61
12. Columns 63-67
13. Columns 69-72
14. Column 74
15. Columns 77-82
16. Columns 84-90
17. Columns 91-97
18. Columns 98-104
19. Columns 106-111
20. Columns 113-119
21. Columns 120-126

Tap bus number (I) \* (the non-unity tap side of the model for transformers or phase shifters)  
Z bus number (I) \* (the side of the model the device impedance is on for transformers and phase shifters)  
Load flow area (I)  
Loss zone (I)  
Circuit (I) \* (Use 1 for single lines)  
Type (I) \*  
0 - Transmission line  
1 - Fixed tap  
2 - Variable tap for voltage control  
3 - Variable tap (turns ratio) for MVAR control  
4 - Variable phase angle for MW control (phase shifter)  
Branch resistance R, per unit (F) \*  
Branch reactance X, per unit (F) \* No zero impedance lines  
Line charging B, per unit (F) \* (total line charging, +B)  
Line MVA rating No 1 (I)  
Line MVA rating No 2 (I)  
Line MVA rating No 3 (I)  
Control bus number  
Side (I)  
0 - Controlled bus is one of the terminals  
1 - Controlled bus is near the tap side  
2 - Controlled bus is near the impedance side (Z bus)  
Transformer final turns ratio (F)  
Transformer (phase shifter) final angle (F)  
Minimum tap or phase shift (F)  
Maximum tap or phase shift (F)  
Step size (F)  
Minimum voltage, MVAR or MW limit (F)  
Maximum voltage, MVAR or MW limit (F)

**Section end line:**

1. Columns 1- 4
- "-999"

# Appendix G

## Load Flow Input Data for a 9 Bus Test System

08/20/97 9BusTestStn									
The9BusSystem.rdf									
BUS DATA FOLLOWS									
#	Type	P1	Q1	Pg	V	Vmin	Vmax	Pgmin	Pgmax
1	3	0.000	0.000	0.000	1.050	0.850	1.050	0.000	2.250
2	2	0.000	0.000	1.250	1.000	0.850	1.050	0.250	1.900
3	2	0.000	0.000	1.250	1.000	0.850	1.050	0.200	1.250
4	1	0.000	0.000	0.000	1.000	0.850	1.050	0.000	0.000
5	1	1.250	0.500	0.000	1.000	0.850	1.050	0.000	0.000
6	1	0.900	0.300	0.000	1.000	0.850	1.050	0.000	0.000
7	1	0.000	0.000	0.000	1.000	0.850	1.050	0.000	0.000
8	1	1.000	0.350	0.000	1.000	0.850	1.050	0.000	0.000
9	1	0.000	0.000	0.000	1.000	0.850	1.050	0.000	0.000
-999 *									
BRANCH DATA FOLLOWS									
#	TBus	ZBus	Type	R	X	B/2	Tap	Rating	ITEMS
1	1	4	1	0.0000	0.0576	0.000	1.000	3.000	
2	4	5	0	0.0100	0.0850	0.088	0.000	3.000	
3	4	6	0	0.0170	0.0920	0.079	0.000	2.000	
4	5	7	0	0.0320	0.1610	0.153	0.000	2.000	
5	6	9	0	0.0390	0.1700	0.179	0.000	2.000	
6	7	2	1	0.0000	0.0625	0.000	1.000	2.000	
7	7	8	0	0.0085	0.0720	.0745	0.000	2.000	
8	9	3	1	0.0000	0.0586	0.000	1.000	1.500	
9	9	8	0	0.0119	0.1008	.1045	0.000	3.000	
-999 *									

Figure G.1. Power flow input data of the nine bus power system.

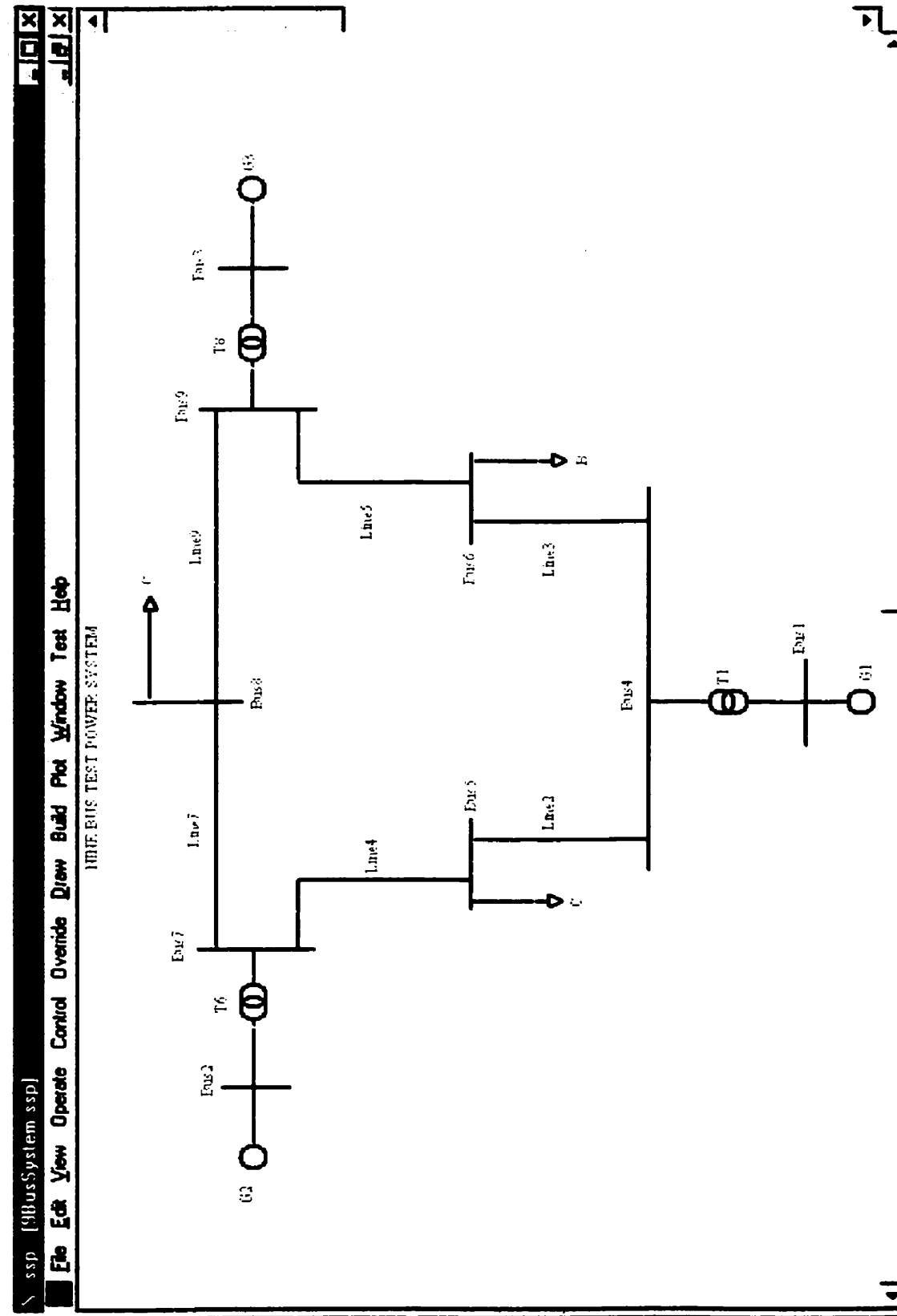


Figure G.2. Single line diagram of a nine bus power system.

## **Appendix H**

### **Test Results**

A small sample of test results were provided in Chapter 7. More test results are provided in this appendix.

#### **H.1. Interlock Tests**

2. Close isolator 4, as shown in Figure H.1.
- Operating state: Normal operating state of the substation.
- Result: The SSP refuses to close isolator 4, and displays the message shown in Figure H.2. The reason for not closing isolator 4 are as follows:
- Isolator 3, which is parallel to isolator 4 is already closed.
  - Circuit breaker B1 is in closed state.
  - There is no short circuit path, which could have been created via switches 1, B5 and 2, across isolator 4.
3. Close isolator 4, as shown in Figure H.3.
- Operating state: Normal operating state of the substation, but now Bus1 and Bus2 are connected to each other through a bus-coupler circuit, which comprises of switches 1, B5 and 2.
- Result: The SSP closes isolator 4 as shown in Figure H.4. The reason for closing isolator 4 is as follows:
- There is now a short circuit path across isolator 4, which is created via switches 1, B5 and 2, across isolator 4.
4. Open isolator 2, as shown in Figure H.5.
- Operating state: Normal operating state of the substation, but Bus1 and Bus2 are connected to each other through a bus-coupler circuit, which comprises of switches 1, B5 and 2.

<b>Result:</b>	<p>The SSP refuses to open isolator 2 and displays the message shown in Figure H.6. The reasons for not opening isolator 2 are as follows:</p> <ul style="list-style-type: none"> <li>• The adjacent circuit breaker B5, which is contained in the same branch as isolator 2, is not open.</li> </ul>
<b>5.</b>	Open isolator 4, as shown in Figure H.7.
<b>Operating state:</b>	The operating state that resulted after performing Test 3.
<b>Result:</b>	<p>The SSP opens isolator 5 as shown in Figure H.8. The reasons for opening the isolator 4 is as follows:</p> <ul style="list-style-type: none"> <li>• There is a short circuit path across isolator 4 via the bus coupler circuit that comprises of switches 1, B5 and 2.</li> </ul>
<b>8.</b>	Open isolator 13, as shown in Figure H.9.
<b>Operating state:</b>	The operating state that resulted after performing Test 7.
<b>Result:</b>	<p>The SSP opens isolator 13 as shown in Figure H.10. The reasons for opening the isolator are follows:</p> <ul style="list-style-type: none"> <li>• Circuit breaker B3 adjoining isolator 13 is open.</li> <li>• Isolator 13 does not interrupt any load or fault current.</li> </ul>
<b>9.</b>	Close isolator 8 in Substation 2, as shown in Figure H.11.
<b>Operating state:</b>	Normal operating state of the substation. Circuit parallel to isolator 8 is feeding power supply to Load1. Circuit breaker B3, can isolate the circuit if a fault appears anywhere in Load1.
<b>Result:</b>	<p>The SSP does not close isolator 8 but displays a message as shown in Figure H.12. The reasons for not closing the isolator are follows:</p> <ul style="list-style-type: none"> <li>• There is no short circuit path across isolator 8.</li> <li>• The branch parallel to the isolator is carrying current.</li> </ul>
<b>10.</b>	Open isolator 13, as shown in Figure H.13.
<b>Operating state:</b>	Normal operating state of the substation. The branch containing isolator 13 also contains circuit breaker B3 which closed and the branch is feeding Load1.
<b>Result:</b>	The SSP does not open isolator 13 but displays message as shown in Figure H.14. The reasons for not opening the isolator are as

follows:

- Circuit breaker B3, adjoining isolator 13, is in closed state.
- Isolator 13 interrupts the load current flowing to Load1 if the isolator were to be opened.

11. Open circuit breaker B2, as shown in Figure H.15.
- Operating state: Normal operating state of the substation. The branch containing circuit breaker B2 connects the Source 2 with the rest of the circuits.
- Result: The SSP does not open circuit breaker B2 but displays message as shown in Figure H.16. The reason for not opening the circuit breaker is as follows:
- Circuit breaker B2 would interrupt current flowing through Source 2.
- Assume that the operator chose to cancel the operation and chose to provide an alternate connection between Source 2 and the rest of the lines.
12. Close isolator 1, as shown in Figure H.17. (It is intended to energize Bus2 through the bus coupler circuit containing isolator 1.)
- Operating state: Normal operating state of the substation, that prevailed after selecting option “No” in the previous test. Bus1 and Bus2 are disconnected.
- Result: The SSP closes isolator 1 as shown in Figure H.18 (where a command to operate isolator 2 is being issued). The reasons for closing the isolator are as follows:
- Adjoining circuit breaker B5 is open.
  - The isolator does not make or break currents.
13. Close isolator 2, as shown in Figure H.18.
- Operating state: Operating state of the substation that resulted after performing Test 12. Bus1 and Bus2 are still disconnected.
- Result: The SSP closes isolator 1 as shown in Figure H.19 (where a command to operate circuit breaker B5 is being issued). The reasons for closing the isolator are as follows:

- Adjoining circuit breaker B5 is open.
  - The isolator does not make or break currents.
14. Close circuit breaker B5, as shown in Figure H.19.
- Operating state: Operating state of the substation that resulted after performing Test 13. Bus1 and Bus2 are still disconnected.
- Result: The SSP closes circuit breaker B5 as shown in Figure H.20 (where a command to operate isolator 6 is being issued). The reasons for closing the circuit breaker are as follows:
- It does not overload any circuit.
  - It does not create any ground fault.
- This operation energizes Bus2 and it is safe to energize Bus2 in this manner because circuit breaker B5 could isolate Bus2 again in case a fault is encountered in Bus2.
15. Close isolator 6, as shown in Figure H.20.
- Operating state: Operating state of the substation that resulted after performing Test 14. Bus2 is now energized.
- Result: The SSP closes isolator 6 as shown in Figure H.21 (where a command to open circuit breaker B2 is being issued). The reasons for closing the isolator are as follows:
- It has a short circuit path across it via switches 12, B2, 5, 1, B5, and 2.
  - The short circuit path includes its affiliated circuit breaker B5.
- This operation has now created a short circuit path across the branch containing the switches 12, B2 and 5. Source 2 is, therefore, connected to Bus1 via two alternate routes.
16. Open circuit breaker B2, as shown in Figure H.21.
- Operating state: Operating state of the substation that resulted after performing Test 15. Source 2 is now connected to Bus1 through two parallel paths.
- Result: The SSP opens circuit breaker B2 as shown in Figure H.22 (where a command to open isolator 5 is being issued). The reasons for



closing the circuit breaker are as follows:

- It does not break the connection between Source 2 and the rest of the lines.
- It does not cause overloading of lines.

The circuit breaker can now be safely isolated by opening isolators 5 and 12 as demonstrated in the next two tests.

17. Open isolator 5, as shown in Figure H.22.

Operating state: Operating state of the substation that resulted after performing Test 16. Branch containing the isolator is now open.

Result: The SSP opens isolator 5 as shown in Figure H.23 (where a command to open isolator 12 is being issued). The reasons for closing the isolator are as follows:

- It does not interrupt currents in the circuit.
- Its adjoining circuit breaker B2 is open.

18. Open isolator 12, as shown in Figure H.23.

Operating state: Operating state of the substation that resulted after performing Test 17. Branch containing the isolator is still open.

Result: The SSP opens isolator 12 as shown in Figure H.24. The reasons for closing the isolator are as follows:

- It does not interrupt currents in the circuit.
- Its adjoining circuit breaker B2 is open.

The circuit breaker B2 is now fully isolated. Substation personnel can now perform maintenance work on B2 after appropriately grounding it.

20. Open isolator I43, as shown in Figure H.25.

Operating state: Normal operating state of the substation.

Result: The SSP refuses to open I43 and displays the message shown in Figure H.26. The reason for not opening the switch is as follows:

- Circuit breaker C4 is closed, when its status tells that it can be opened. When C4 is “healthy”, it must be opened first before opening isolator I43.

Now it is found that the grounding switch will not open till all three isolators around it are open and the isolators would not open unless the circuit breaker C4 is open.

21. Open circuit breaker C4, as shown in Figure H.27.
- Operating state: Normal operating state of the substation.
- Result: The SSP does not open the circuit breaker C4 but displays a warning message as shown in Figure H.28. The SSP warned because it found that the opening of C4 will interrupt the load current flowing through line 7L4.

At this instance, the SSP had given two choices, either to proceed ahead and open the circuit breaker or to back off and not open it. For we are going to test the effect of opening the circuit breaker on the status of its adjoining switches, the “Yes” button to permit the SSP to open the switch was selected.

22. Open isolator I43, as shown in Figure H.29.
- Operating state: Operating state resulting after the last test. The circuit breaker C4 has been opened by selecting “Yes” in the dialog box shown in Figure H.29. It is apparent from the figure that the opening of circuit breaker C4 has caused the isolators, I41, I43 and I44 to be enabled.
- Result: The SSP opens isolator I43. Figure H.30 shows the resulting status of the switches, where a command to open another isolator is being issued. The reason for not operating I43 are as follows:

- Grounding switch G42 is open and circuit breaker C4 is open. It is, therefore, not locked by the interlock expression that blocked the operation during Test 20.
- It does not interrupt currents and does not alter the voltages or currents in the circuit.

The status of grounding switches G41 and G42 indicate that they are still disabled.

23. Open isolator I41, as shown in Figure H.30.

<b>Operating state:</b>	Operating state resulting after Test 22. The switches C4 and I43 are now open. The isolator I41 is enabled since the circuit breaker C4 was opened..
<b>Result:</b>	<p>The SSP closes isolator I41. The resulting state of the substation is shown in Figure H.31. The reasons for opening the isolator are as follows:</p> <ul style="list-style-type: none"> <li>• Isolator I42, grounding switch G41 and circuit breaker C4 are open. The isolator is, therefore, not locked by its interlock expression.</li> <li>• Isolator I41 does not interrupt currents and does not alter the voltages or currents in the circuit.</li> </ul> <p>The status of grounding switches G41 and G42 indicate that they are now free to operate. This is because the circuit breaker C4 is open and isolators I41, I42 and I43 are open.</p>
<b>24.</b>	Close grounding switch G42, as shown in Figure H.32.
<b>Operating state:</b>	That resulted after performing Test 23. The grounding switches G41 and G42 are free to operate.
<b>Result:</b>	<p>The SSP closes grounding switch G42 as shown in Figure H.33. This was possible because of the following reasons:</p> <ul style="list-style-type: none"> <li>• The circuit being grounded is fully isolated from the rest of the circuit.</li> </ul> <p>The switches I43 and C4 are now disabled. The SSP will not close them unless the grounding switch G42 is opened again.</p>
<b>25.</b>	Close grounding switch G41, as shown in Figure H.34.
<b>Operating state:</b>	That resulted after performing Test 24. The grounding switches G42 is closed and G41 is free to operate.
<b>Result:</b>	<p>The SSP closes grounding switch G41 as shown in Figure H.35. This was possible because of the following reasons:</p> <ul style="list-style-type: none"> <li>• The circuit being grounded is fully isolated from the rest of the circuit.</li> </ul> <p>The switches I41, I42, I43 and C4 are now disabled. The SSP will not close them unless the grounding switches are opened again.</p>

The switches C4, I41 and I43 again become enabled, when G41 and G43 are opened.

While performing these tests, isolator I42 remained disabled except when isolator I41 was open.

26. Close bus coupler C0, as shown in Figure H.36.

Operating state: Normal operating state except that isolators I01 and I02 of the bus coupler circuit are closed. Isolators I12, I22, I32, I42, I52 are disabled as the isolator adjacent to them and connected to Bus1 are closed.

Result: The SSP closes C0, as shown in Figure H.37. This was possible because of the following reason:

- The closing of the circuit breaker does not alter the flow of current in system.

The switches I41, I42, I43 and C4 are now enabled due to the following reason:

- There is a short circuit path across them via the bus coupler circuit.

## **H.2. Switching Sequence Test**

A small sample of the results of switching sequence tests were provided in Chapter 7. Some more test results are provided in this appendix.

27: Disconnect Load1 from busbars, as shown in Figure H.38.

Operating state: Normal. All lines are connected to Bus1. The auxiliary bus Bus2 is not energized.

Result: The SSP disconnects Load1 from Bus1 as shown in Figure H.39 by opening the following switches:

Circuit breaker B3

Isolator 7

In this case, the circuit breaker has been opened before the isolator so that the current flowing through the circuit between Load1 and

**Bus1 is already interrupted before the isolator starts opening.**

- 28:** Disconnect Source1 from busbars, as shown in Figure H.40.
- Operating state:** The state that resulted after executing Test 27. Load1 is now disconnected from the busbars.
- Result:** The SSP disconnects Source1 from Bus1 as shown in Figure H.41 by opening the following switches:
- Circuit breaker B1**
  - Isolator 3**
- Again in this case, the circuit breaker has been opened before the isolator.
- 
- 29:** Connect Source1 to Load1, as shown in Figure H.42.
- Operating state:** The state that resulted after executing Test 28. Lines Load1 and Source1 are both disconnected from the busbars.
- Result:** The SSP connects Source1 with Load1 as shown in Figure H.43 by opening the following switches:
- Isolator 7**
  - Isolator 3**
  - Circuit breaker B1**
  - Circuit breaker B3**
- In this case, the following events are taking place:
- The isolators have been closed before the circuit breakers ensuring that the isolators do not make any current.
  - When the circuit breakers are closed, the source side breaker is closed before the load side breaker.
  - The lines are connected via Bus1 not Bus2. The SSP has chosen an already energized (and also having higher priority) bus in establishing the connections.
- 
- 30:** A. Disconnect Load1 from bus, as shown in Figure H.44.  
B. Disconnect Load3 from busbars, similarly as done for Load1.
- Operating state:** Normal operating state. This is Test Substation 3 as shown in Figure 7.3.

**Result:** A. The SSP disconnects Load1 from Bus1 by opening the following switches:

- Circuit breaker B3
- Isolator 7

B. The SSP disconnects Load3 from Bus3 and Bus4 by opening the following switches:

- Circuit breaker B8
- Circuit breaker B12

The final state of the switches are shown in Figure H.45, which also shows the event log that records the order of operations.

**31:** A. Connect Load1 to bus, as shown in Figure H.46.  
 B. Connect Load3 to bus, as done for Load1.

**Operating state:** The state that resulted after performing Test 30.

**Result:** A. The SSP connects Load1 to Bus1 by closing the following switches:

- Isolator 7
- Circuit breaker B3

B. The SSP connects Load3 to Bus3 and Bus4 by closing the following switches:

- Circuit breaker B8
- Circuit breaker B12

The final state of the switches are shown in Figure H.47, which also shows the event log that records the order of operations. The SSP automatically recognizes the differences in the configuration of the sections of the busbars. It connects the load to both the busbars when a circuit breaker was available on paths leading to each busbar. The isolators have been closed before the circuit breakers when connecting a line to a busbar.

**32:** A. Disconnect Load2 from busbar, similarly as done in earlier tests.  
 B. Declare isolator 9 as unhealthy as shown in Figure H.48.  
 C. Connect Load2 to busbar as shown in Figure H.49.

**Operating state:** Operating state that resulted after performing Test 31.

**Result:**

**A. The SSP disconnects Load2 from Bus1 by opening the following switches:**

- Circuit breaker B4**
- Isolator 9**

**B. The SSP declares isolator 9 as “unhealthy”.**

**C. The SSP connects Load2 to Bus1 by closing the following switches:**

- Isolator 10**
- Isolator 14**
- Isolator 13**
- Circuit breaker B5**
- Circuit breaker B4**

The final state of the switches are shown in Figure H.50, which also shows the event log that records the order of operations. The SSP connected the load to both the energized busbar Bus1 of its own bus-section via Bus2 and the bus-coupler circuit. The SSP closed the isolators involved in the path before the circuit breakers when connecting a line to a busbar.

**33:** Disconnect 7L4 from busbar, as shown in Figure H.51.

**Operating state:** Normal. This is Test Substation 6 as shown in Figure 7.6.

**Result:** The SSP disconnects line 7L4 from the busbars by opening the following switches:

- Circuit breaker C4**
- Isolator I41**
- Isolator I43**
- Isolator I44**

The final state of the switches are shown in Figure H.52, which also shows the event log that records the order of operations.

The following points are to be noted in this operation:

- The SSP opened the circuit breaker involved in the path before opening isolators.

- The SSP opened load side switches first and progressed towards the source side switches.
- The SSP opened the isolators I43 and I44, which it would not do in the absence of G42 and in the absence of branching off the node between I43 and I44.

**35:** Transfer lines from busbar Bus1 to Bus 2, as shown in Figure H.53.

**Operating state:** Normal. All lines of the left section are connected to Bus1 and all lines of the right section are connected to Bus3.

**Result:** The SSP closed the following switches:

Bus coupler isolator 1 and 2

Bus coupler circuit breaker B5

Bus selector switches 4, 6, 10 and 8

Bus sectionalizing isolator 16

Bus coupler isolators 18 and 17

Bus coupler circuit breaker B10

The SSP then opened the following switches:

Bus sectionalizing isolator 15

Bus selector switches 3, 5, 7 and 9

Bus coupler circuit breaker B5

Bus coupler isolators 2 and 1.

The state of the substation after transferring the lines from Bus1 to Bus2 and the switching operations carried during the process are shown in Figure H.54. During this bus transfer process the SSP has performed the following operations:

- Energize Bus2 through bus coupler circuit and bring Bus1 and Bus2 to the same voltage level.
- Create a short circuit path across the bus selector switched.
- Connect all the lines to Bus2 via the bus selector switches adjoining Bus2.
- Connect Bus2 to the energized bus of the next section, which is Bus3, via bus sectionalizing switch.
- Disconnect Bus1 from the adjoining section.
- Prepare lines to disconnect from Bus1 by opening bus



selector switches adjacent to Bus1.

- Disconnect Bus1 from Bus2 by opening the bus coupler circuit between them.

**37:** Isolate the circuit breaker B6, as shown in Figure H.55.

**Operating state:** Normal but it is required to isolate the circuit breaker B6 to perform scheduled maintenance. This is the Test Substation 5 shown in Figure H.5.

**Result:** The SSP closed the following switches:

Bus coupler isolators 17 and 18

Bus coupler circuit breaker B10

Bus transfer isolator 20

The SSP then opened the following switches:

Circuit breaker B6

Isolators I01 and I02, which are adjoining to B6

The state of the substation after isolating the circuit breaker B6 and the switching operations carried during the process are shown in Figure H.56. During this process the SSP has performed the following operations:

- Energize Bus2 through bus coupler circuit
- Connect Source3 to Bus2, and, therefore to Bus1, providing an alternate route of power flow between Source3 and the rest of the lines.
- Interrupt the current flowing through B6
- Isolate B6 by opening the isolators immediately surrounding it.

**39:** Isolate Bus1, as shown in Figure H.57.

**Operating state:** Normal. This is Test Substation 5 as shown in Figure 7.5.

**Result:** The SSP isolated Bus1. The lines connected to Bus1 were first connected to Bus2 by closing the following switches:

Bus coupler switches 1, 2 and B5

Bus selector isolators 4, 6, 10, 8

Bus sectionalizing isolator 16

Bus coupler switches 18, 17 and B10

The SSP then isolates the bus by opening the following switches:

Bus sectionalizing isolator 15

Bus selectors 3, 5, 7, 9

Bus coupler switches B5, 2, and 1

The final state of the substation and the switching operations recorded in the event log are shown in Figure H.58.

**40:** Isolate busbar Bus1, as shown in Figure H.59.

**Operating state:** Normal except that Bus1 needed to be taken out of service for maintenance. This test is carried on Test Substation 7 shown in Figure 7.7.

**Result:** The SSP isolates Bus1 by first connecting all lines to Bus2 by closing the following switches:

Bus coupler isolator I02, I01 and C0

Bus selector isolators I42, I52, I12, I22 and I32

The SSP then opens the following switches to isolate Bus1:

Bus selector isolators I41, I51, I11, I21 and I31

Bus coupler switches C0, I01 and I02

The final state of the substation and the switching operations recorded in the event log are shown in Figure H.60.

**41:** Declare isolator 9 as not healthy, as shown in Figure H.61.

**Operating state:** Normal state of Test Substation 3 shown in Figure 7.3, except that Load2 is disconnected from the busbars.

**Result:** The SSP treated isolator as not healthy. The status of the switch was changed and the isolator 9 was displayed with red color as shown in Figure H.62 where a command is being issued to connect Load2 to a busbar.

**42:** Connect Load2 to a busbar, as shown in Figure H.62.

**Operating state:** State of Test Substation 3 after performing Test 41. Load2 is disconnected from the busbars and isolator 9 is not operable.

**Result:** The SSP connected Load2 to BUS1. The final configuration of the substation and the switches operated during the process are shown

in Figure H.63. The SSP closed the following switches to energize Load2:

Bus selector isolator 10  
Bus coupler isolators 14 and 13  
Bus coupler circuit breaker B5  
Line breaker B4

The SSP performed the following operations during this process.

- Avoided isolator 9 when creating a power flow path
- Connected Load2 to BUS1 that was already energized; it did not simply connect to BUS2 that was not energized
- Chose an energized busbar of its own section than from another section

43:

Issue the following commands in a sequence:

A. Disconnect line 7L4 from busbars; issue the command as shown in Figure H.64 and capture the result in Figure H.65.

B. Issue a command to bypass circuit breaker C4 as shown in Figure H.66.

C. Connect line 7L4 to busbar, as shown in Figure H.67.

Operating state:

Normal All the lines are connected to Bus1.

Result:

A. The SSP disconnects line 7L4 by opening the following switches:

Circuit breaker C4  
Isolators I41, I43 and I44

The state of the substation is shown in Figure H.65.

B. The SSP puts circuit breaker C4 in “bypass” mode to exclude it from new power flow paths. This state change is captured by the event log shown in Figure H.66.

C. The SSP connects line 7L4 to Bus1 while bypassing circuit breaker B4 by closing the following switches:

Isolators I44, I46  
Bus coupler switches I02, I01 and C0

The final state of the substation and the switches operated during the transition are shown in Figure H.68.

### **H.3. Validating voltage and current limits**

The effect of operating a switch on bus voltages and line currents of the system are evaluated by performing load flow calculations. This section presents how the SSP facilitates the incorporation of the load flow data of the system in the validation of switching operations and presentation of line currents and bus voltages for graphical comparison of their values.

A 9 bus system configuration as shown in Figure H.69 is used in this section to illustrate the effects. Test Substation 4, shown in Figure 7.4, is to be represented by bus 3 of the 9 bus system represented the substation. Source2 of the substation is connected to the system bus 9 via transmission line 8. Source1, Load1 and Load2 of the substation are lumped and represented by a single source, 3, in the power system diagram of Figure H.69. System bus 1 is used as a slack bus, where the voltage magnitude is kept constant at 1.05 per unit, and voltage phase angle is considered as 0 degree. This bus injects the active and reactive power not supplied by the rest of the generating units in the system to fulfil the load demand. The phase angles of all other buses are measured with reference to that of the slack bus. Buses 2 and 3 of the 9 bus system are voltage controlled, where the magnitude of the voltage is kept at 1.00 per unit and the amount of active power injected into the buses are kept constant at 1.25 per unit. All other buses are load buses and the active and reactive power drawn from those buses are known.

Figure 1 of Appendix H shows the bus data and line data of the nine bus power system. In the tables, the per unit values are measured on a 100 MVA base. After acquiring the bus data and line data of the power system the substation switching scheme is updated to incorporate the data when making the switching decisions. The SSP has the following interfaces to specify the system data to the switching scheme:

- Edit the single line diagram of the power system as shown in Figure H.69.
- Edit the single line diagram of the substation as shown in Figure H.70 and from the menu select: “Edit” and “Properties” as shown in Figure H.70.
- Specify system bus number to be 3, as shown in Figure H.71. Similarly specify the filenames and directories of the

input and output data files. The minimum and maximum voltage levels within which the bus voltage of the substation is permitted to fluctuate.

- Specify the active and reactive powers at the sources and loads through data dialog boxes as shown in Figure H.72.
- Specify the properties of the input-output nodes that are connected to adjoining substations. These nodes use two extra entry of data to represent the line number of the system that leads to this node and the bus number of the adjoining substation. In the example, Source2 shown in Figure H.72 connects this substation with system bus 9 (Bus9) via line 8.

**44:** A. Run Load Flow as shown in figure H.73. (A command that involves operation of switches such as Open-switch, Close-switch, Connect-line, Disconnect-line and Isolate-component automatically causes the load-flow program to be invoked in the background. This command activates the load flow calculations even when any switching operation is not under way.

B. Plot Voltage, Current as shown in figure H.74.

**Operating state:** Normal operating state of the substation. All lines of Test Substation 4 are connected to Bus1.

**Result:** A. No overloading of line or abnormal bus-voltage is encountered as indicated in Figure 6.75.

B. An application named PlotData is launched and linked with the data-files generated by the load-flow program. The quantity to be plotted are selected from PlotData by using its Quantity menu. The type of graph are selected among Line, 3D-bar and 2D-bar, which is the default graph type. To plot the bus voltage values of all system buses, the selection are made as shown in Figure H.76. The plot for the case under analysis appears as shown in Figure H.77. The numerical value of the data also appear on an edit box beside the graph. The selection made to plot the line current values is shown in Figure H.78; the plot and part of the numeric data file are shown in Figure H.79.

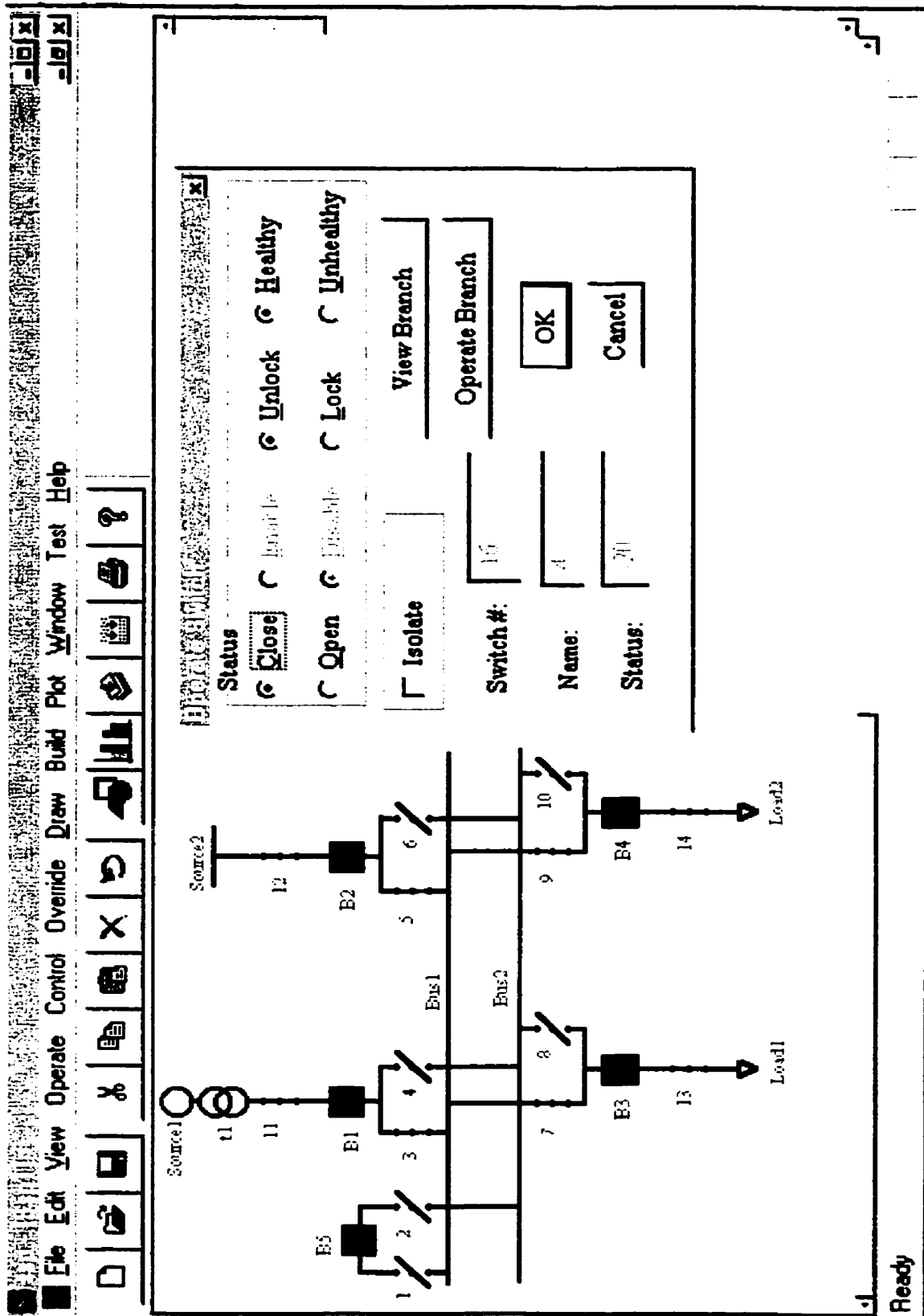


Figure H.1. Command being issued to close isolator 4.

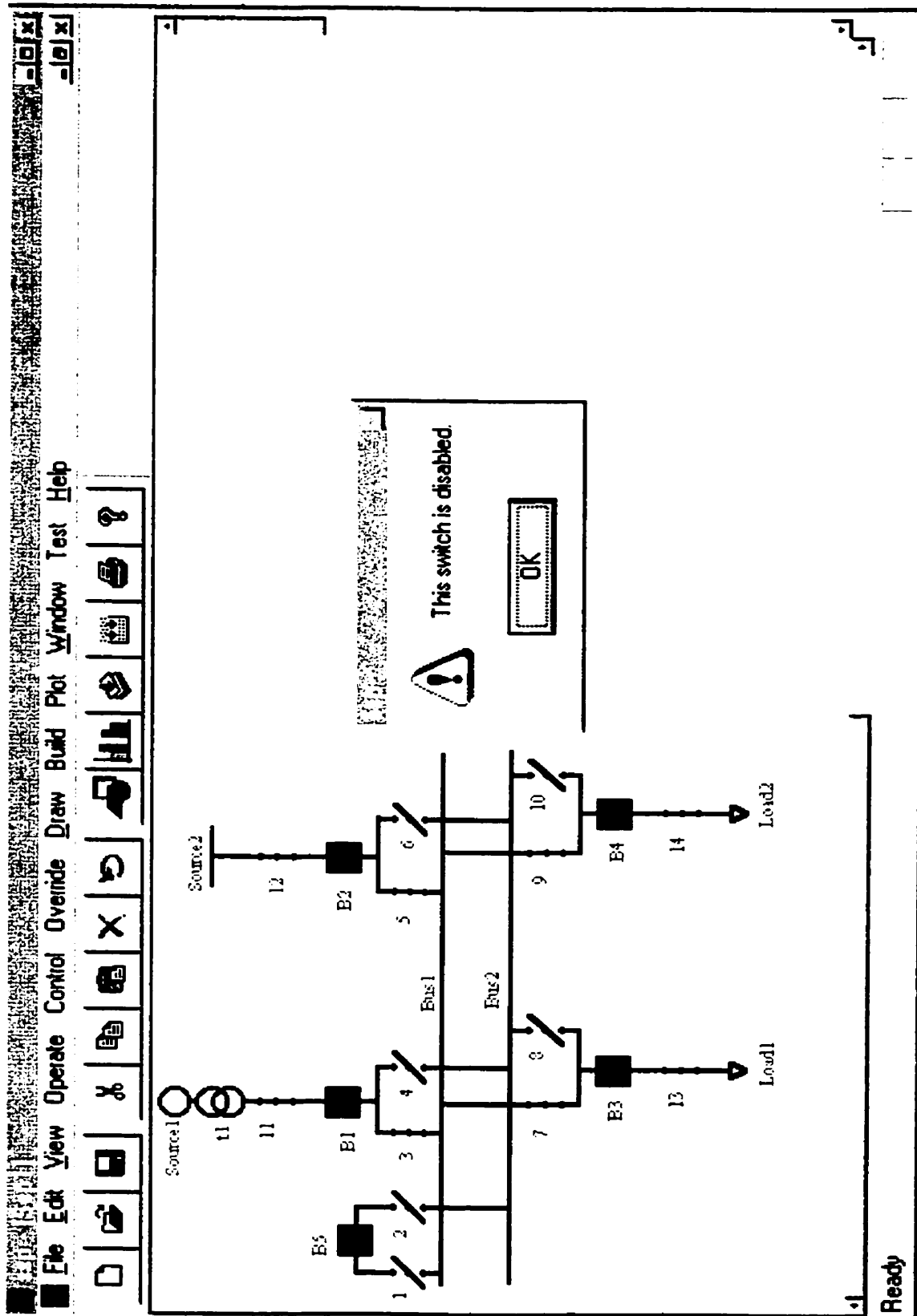
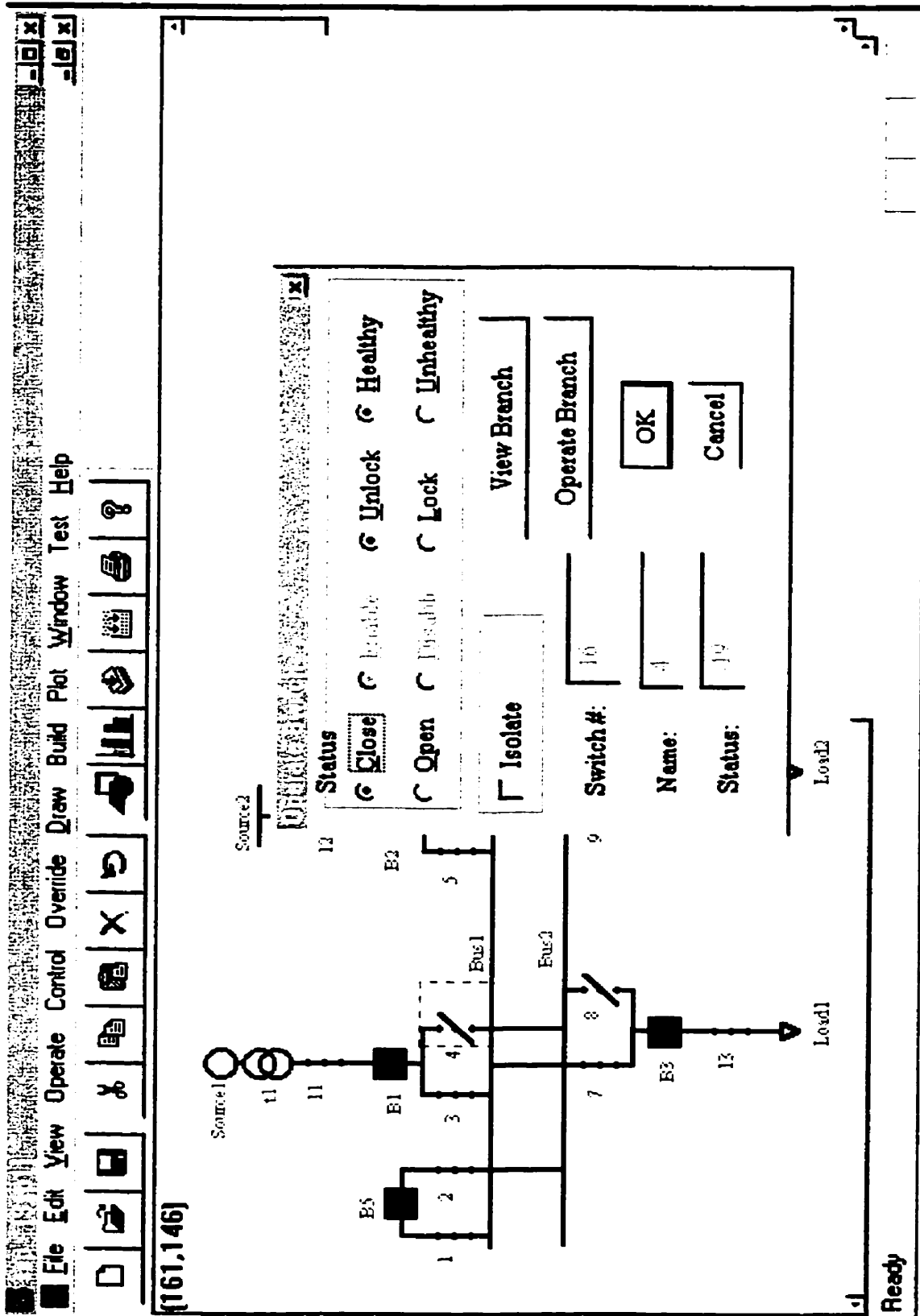


Figure H.2. The SSP refuses to close isolator 4.



**Figure H.3.** Command being issued to open isolator 4 after the bus coupler circuit was closed.



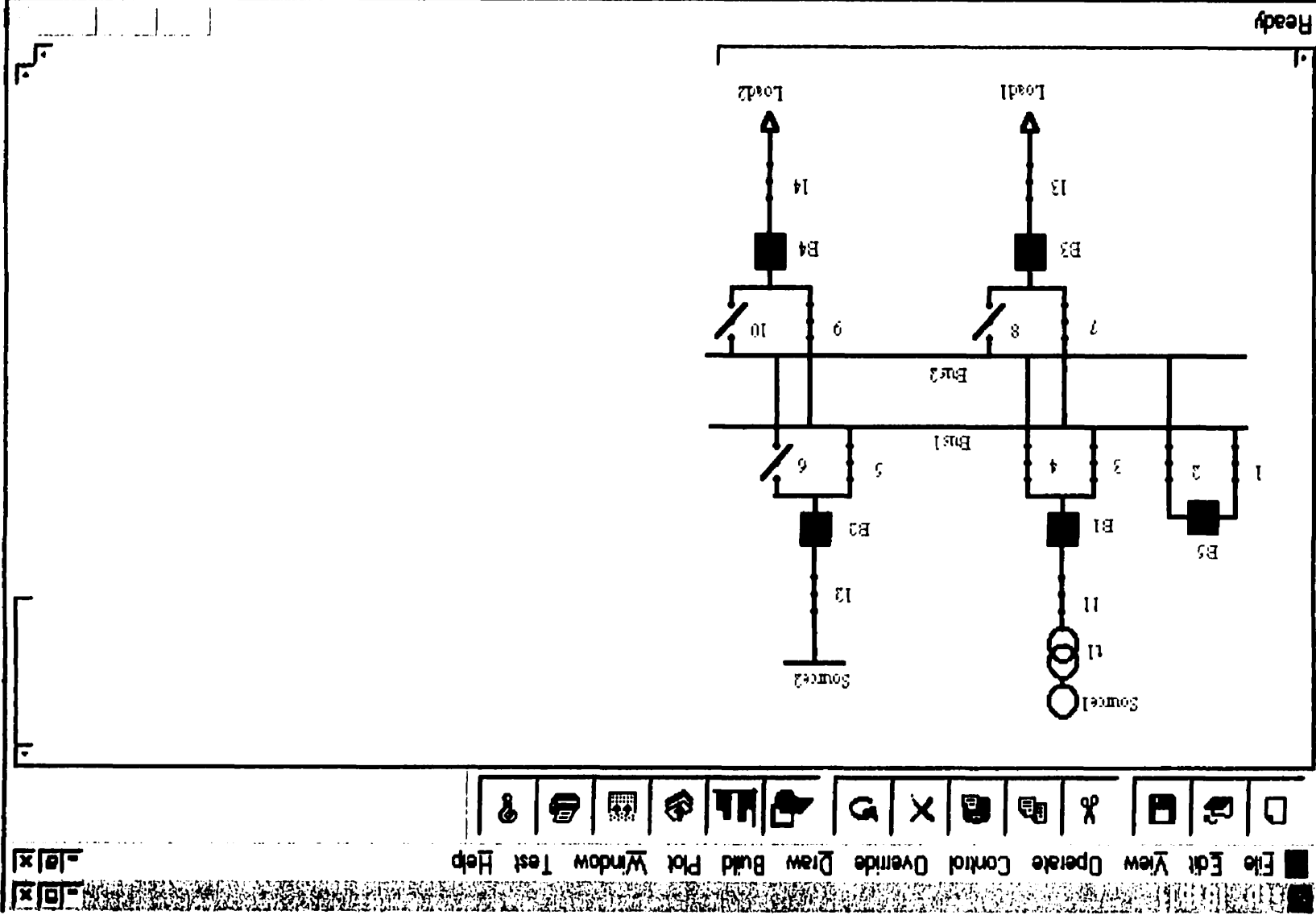


Figure H.4. The SSP closes isolator 4.

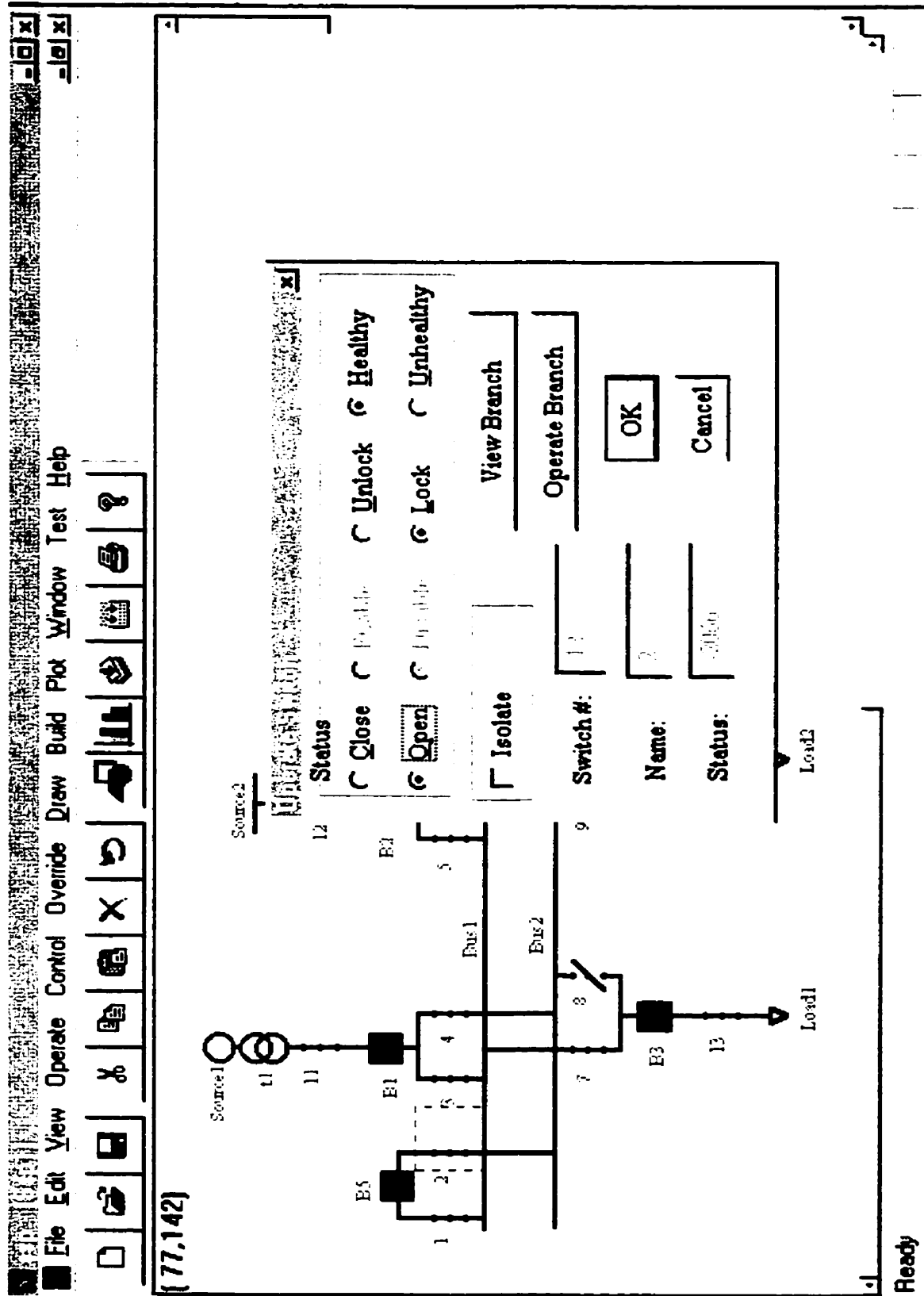


Figure H.5. Command being issued to open isolator 2.

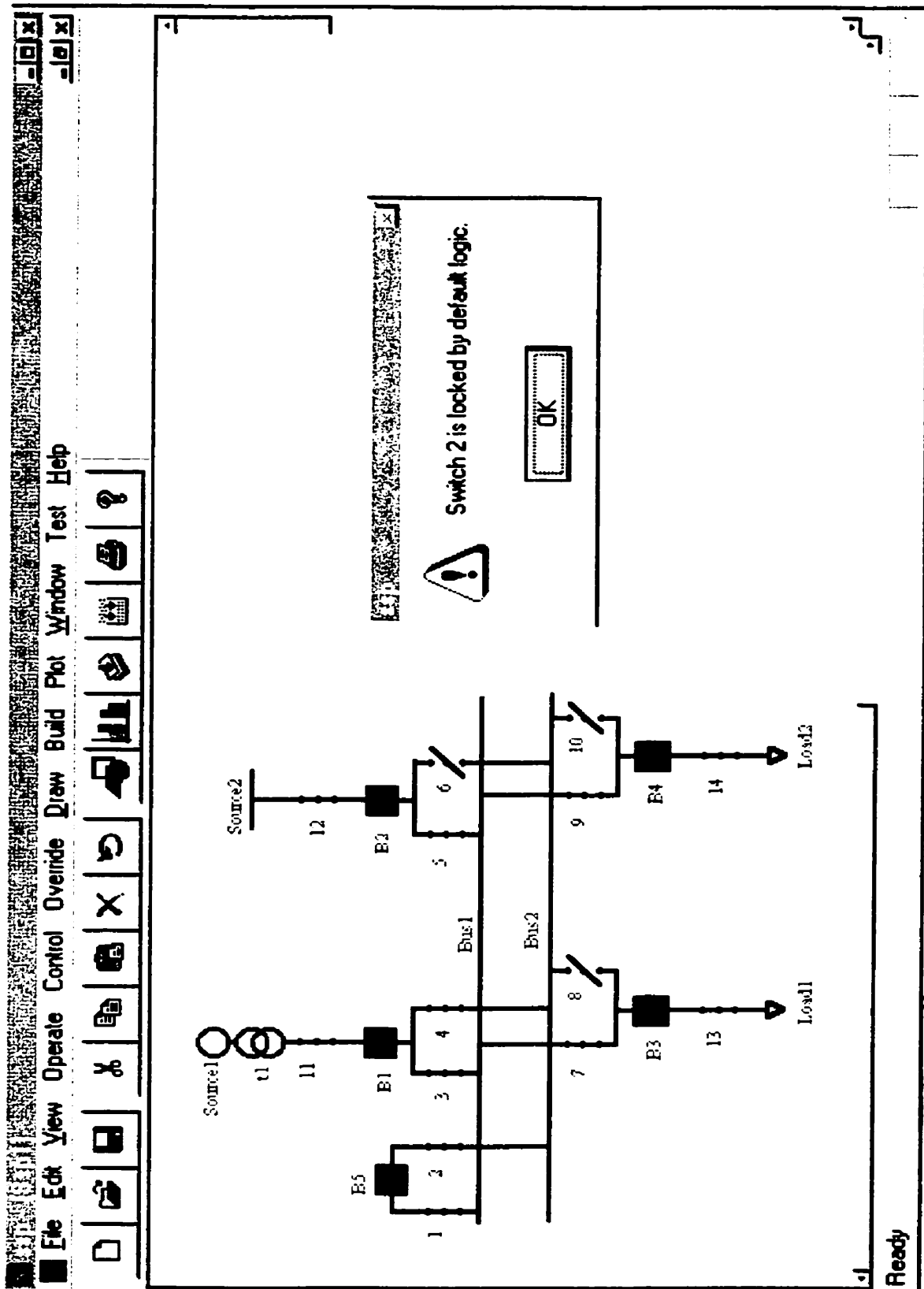


Figure H.6. The SSP refuses to open isolator 2.

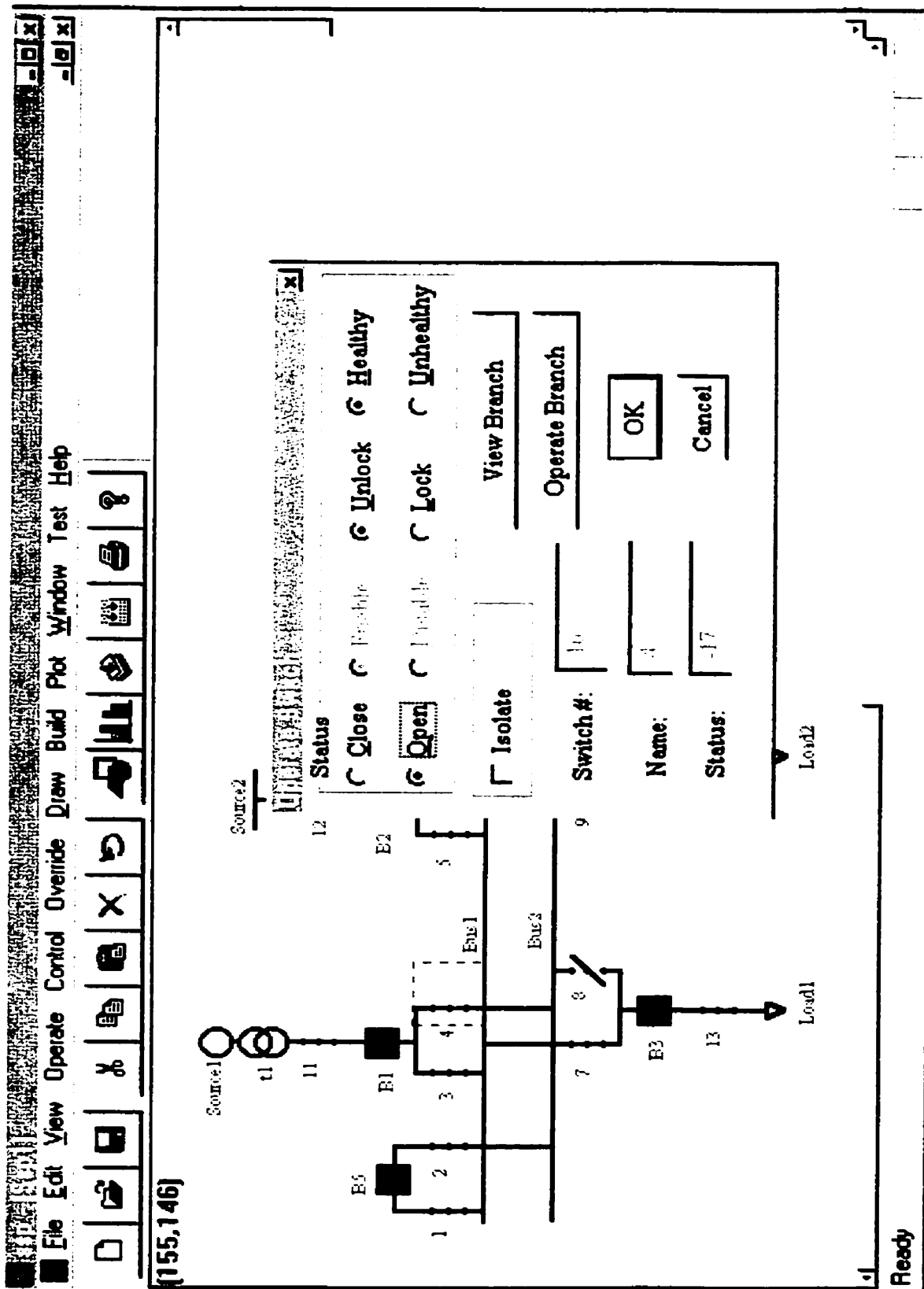


Figure H.7. Command being issued to open isolator 4.

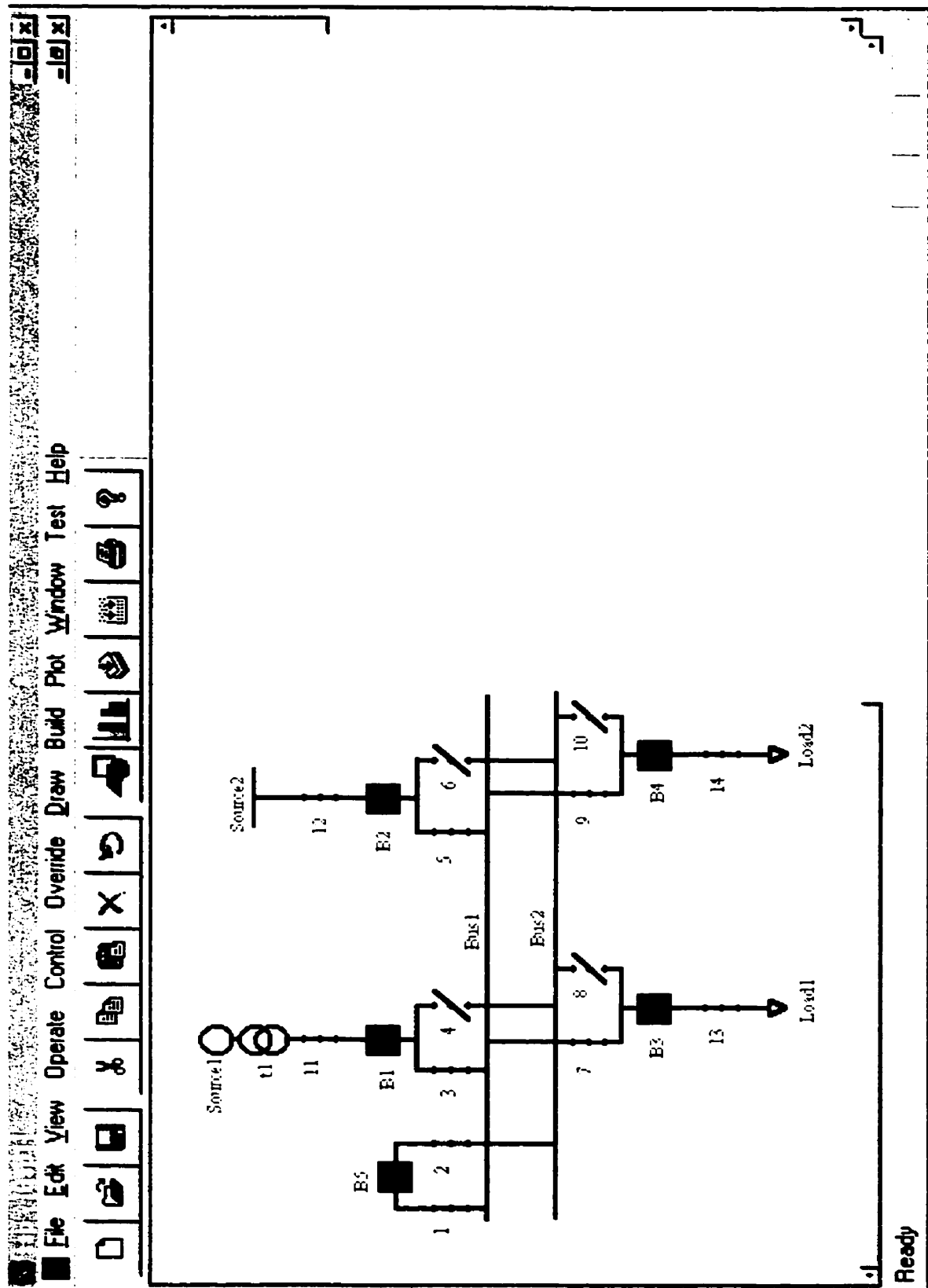


Figure H.8. The SSP opens isolator 4.

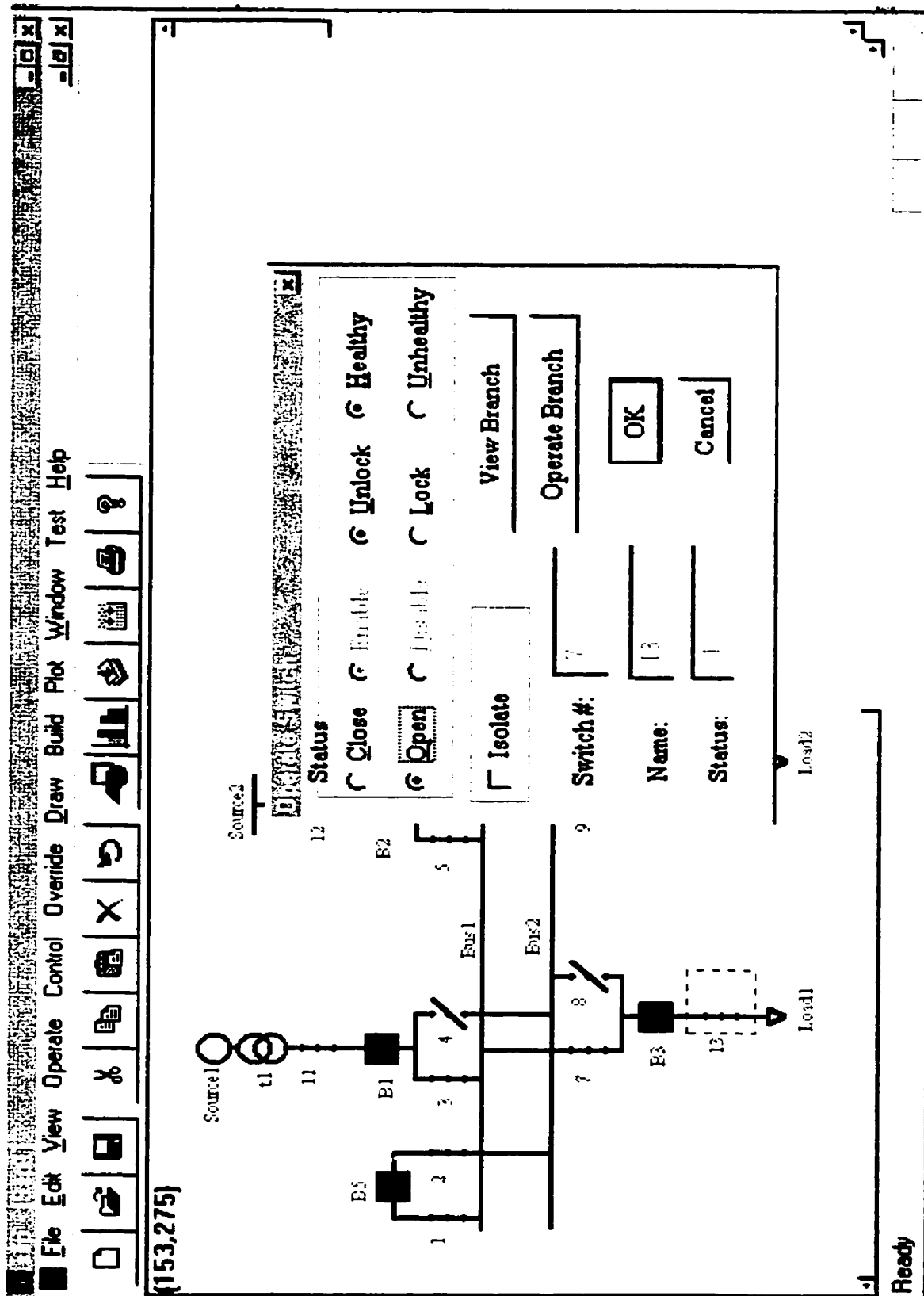


Figure H.9. Command being issued to open isolator 13.

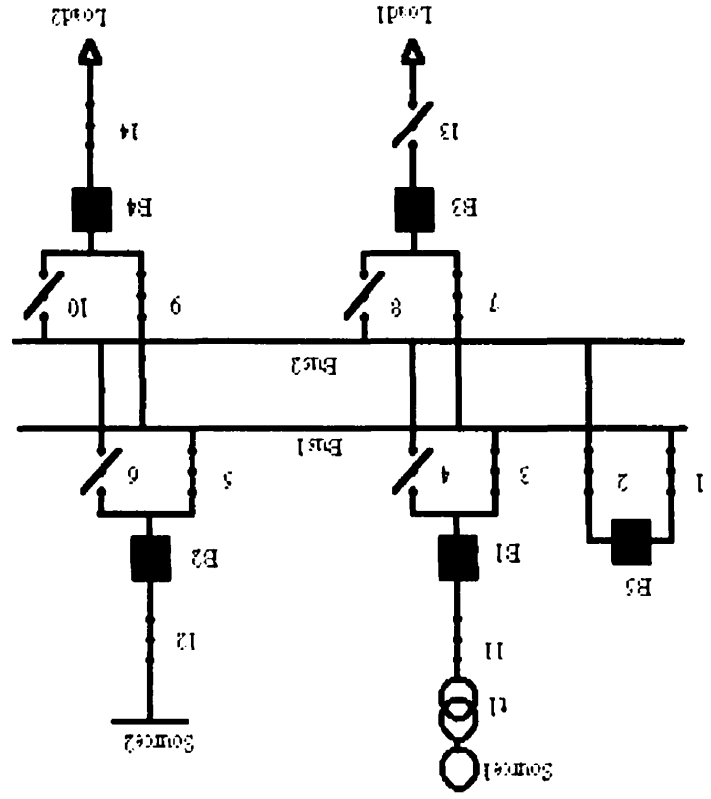


Figure H.10. The SSP opens isolator 13.

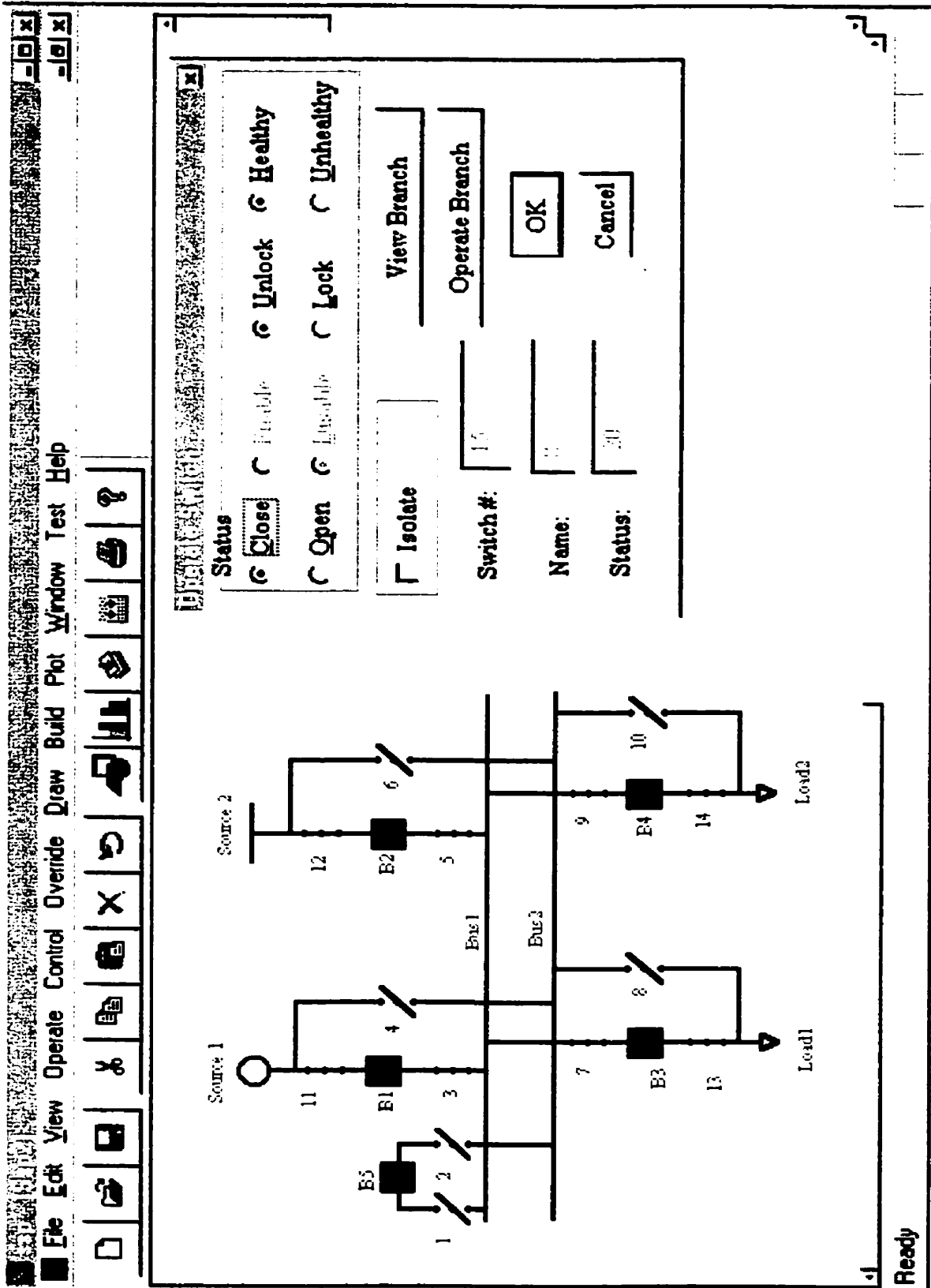


Figure H.11. Command being issued to close isolator 8.



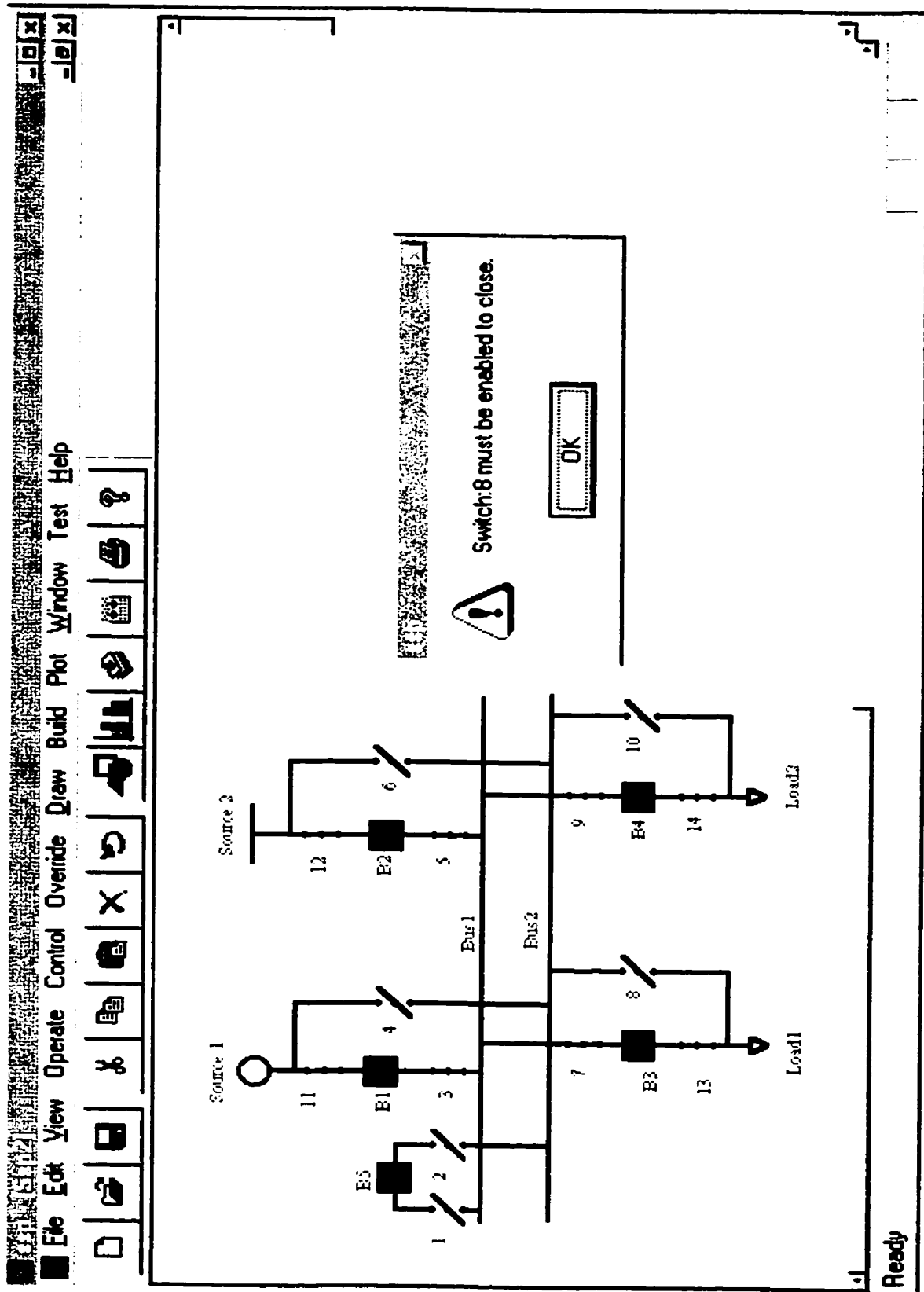


Figure H.12. The SSP refuses to close isolator 8.

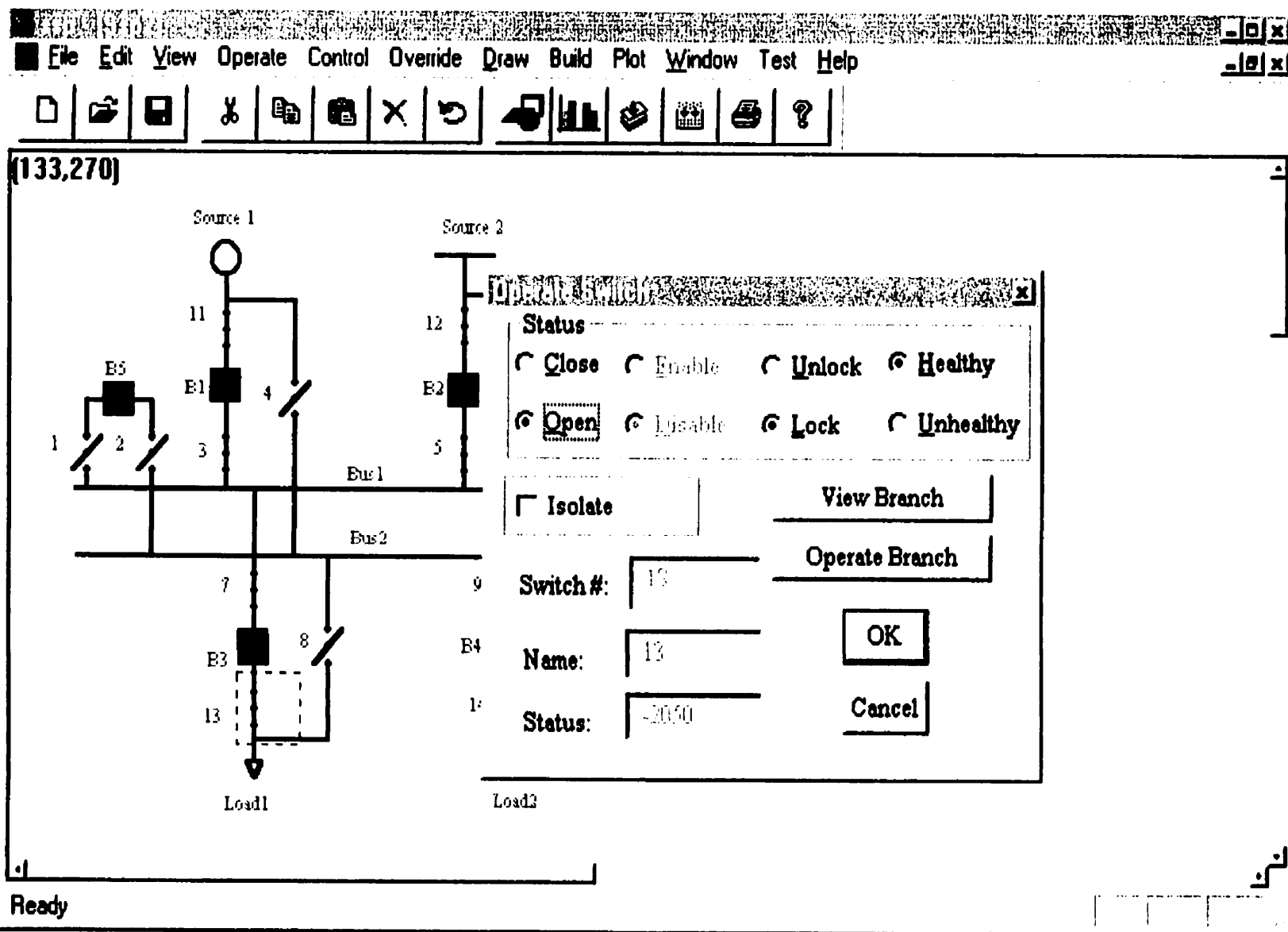


Figure H.13. Command being issued to open isolator 13.

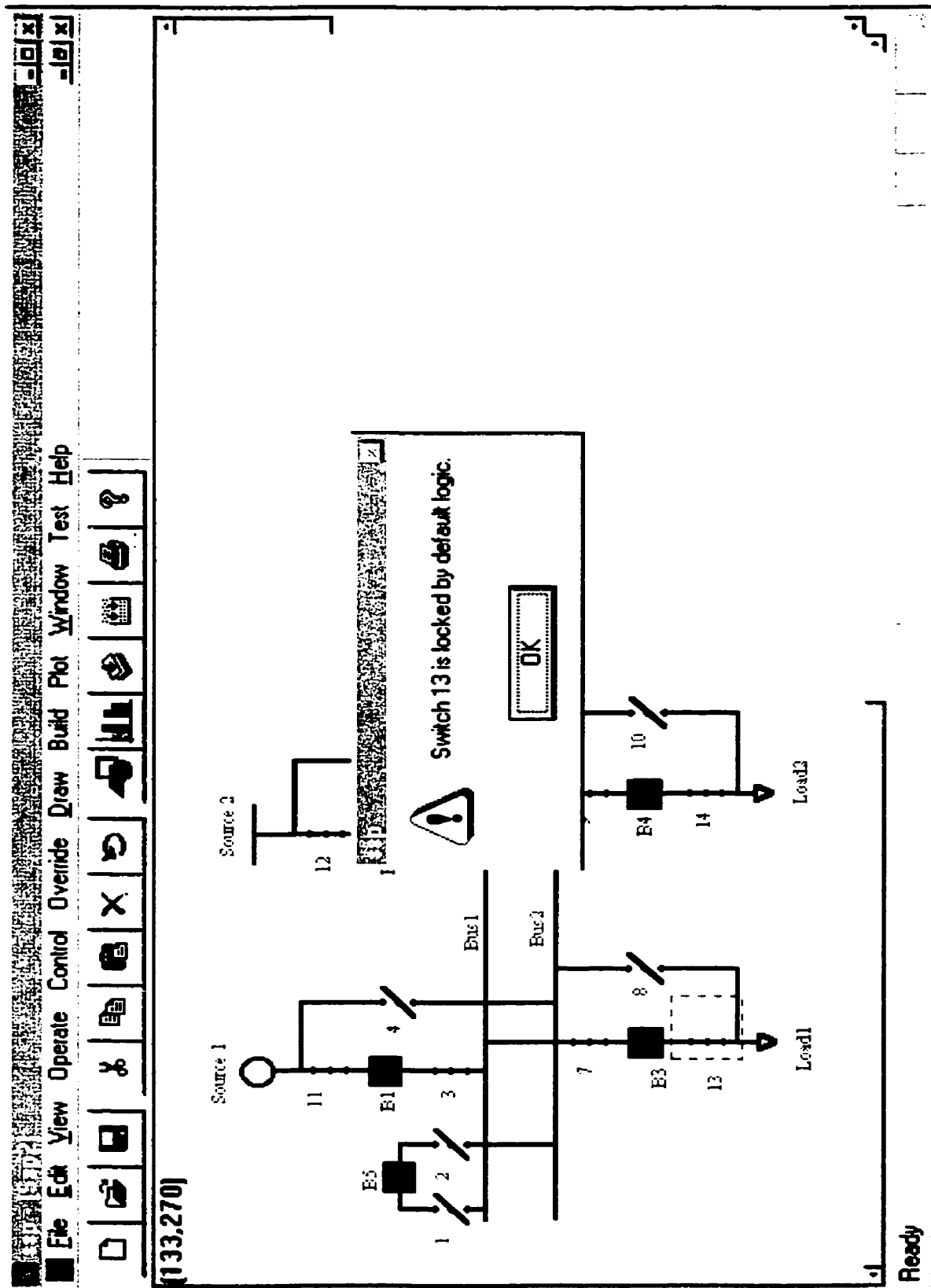


Figure H.14. The SSP refuses to open isolator 13.

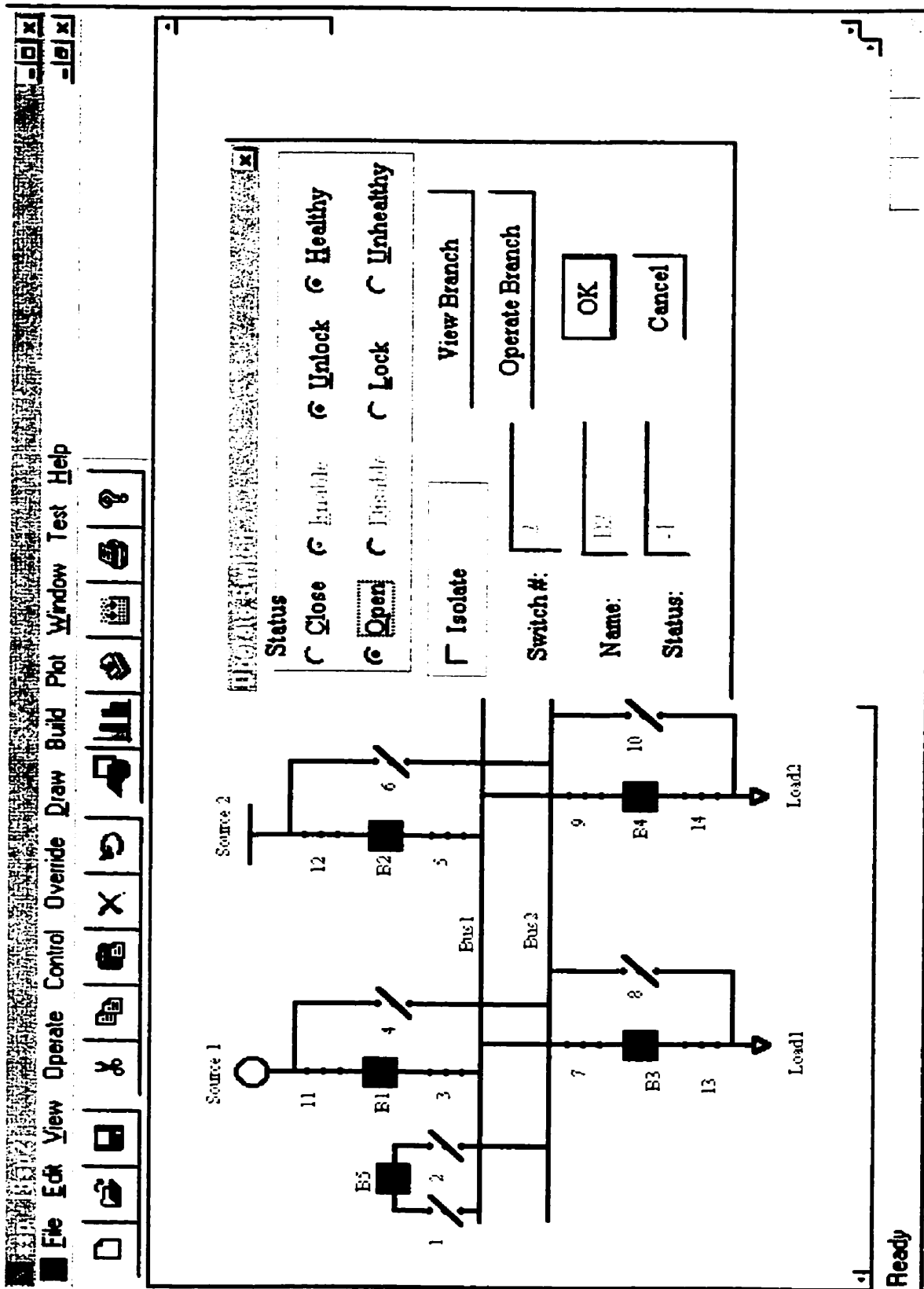
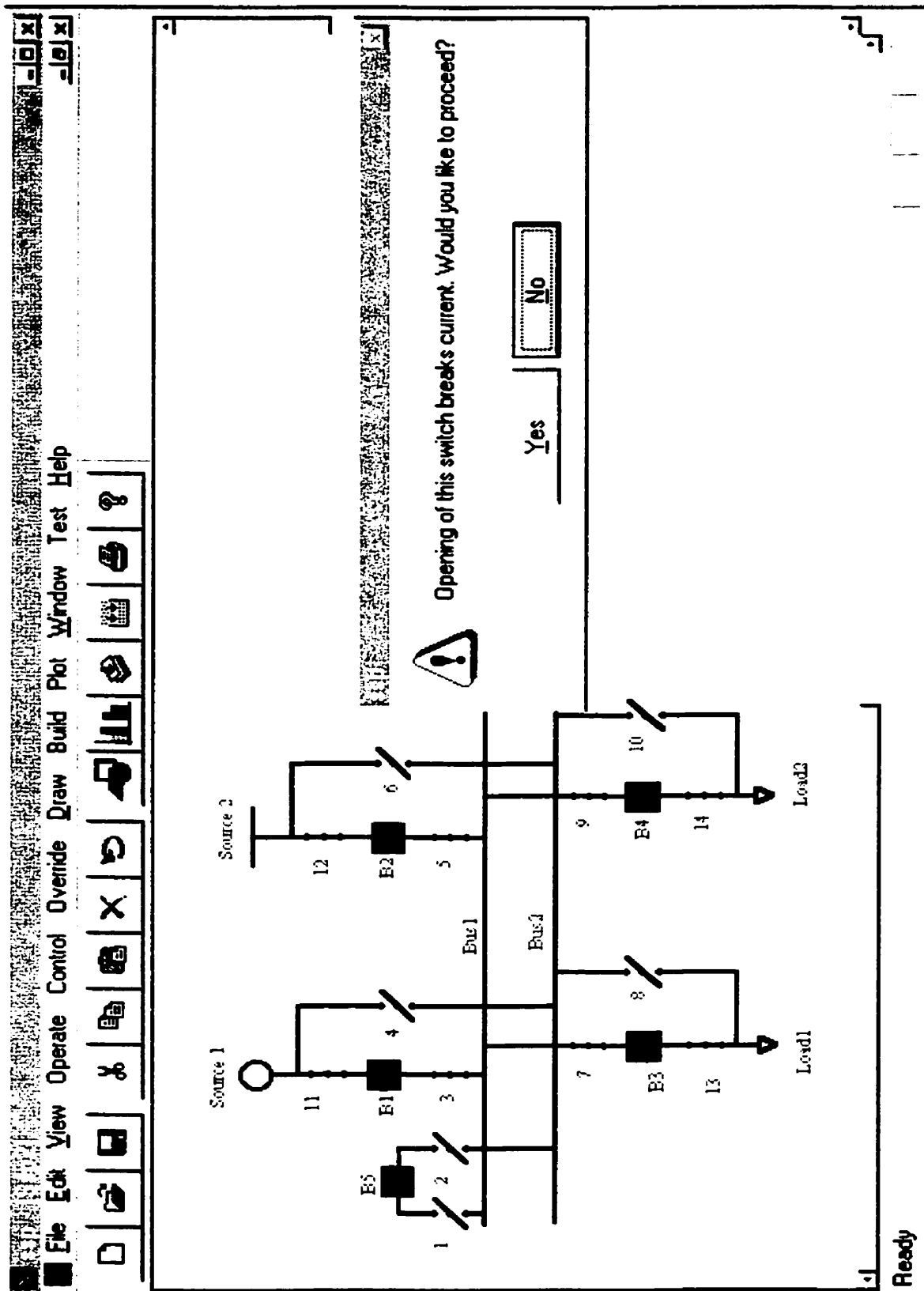


Figure H.15. Command being issued to open circuit breaker B2.



**Figure H.16.** The SSP warns about the interruption of current and gives the operator a chance to cancel the operation.

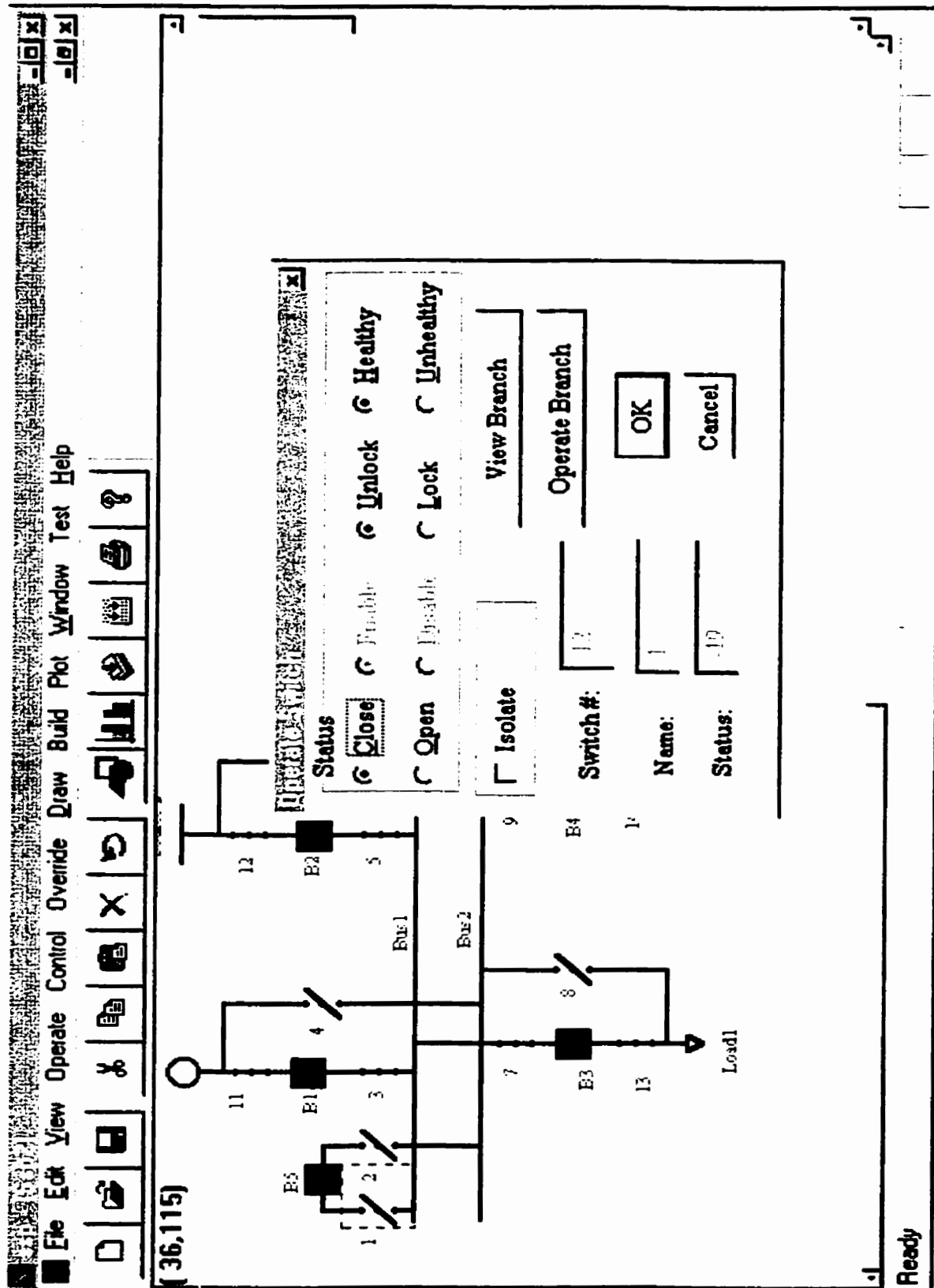


Figure H.17. Command being issued to close isolator 1.

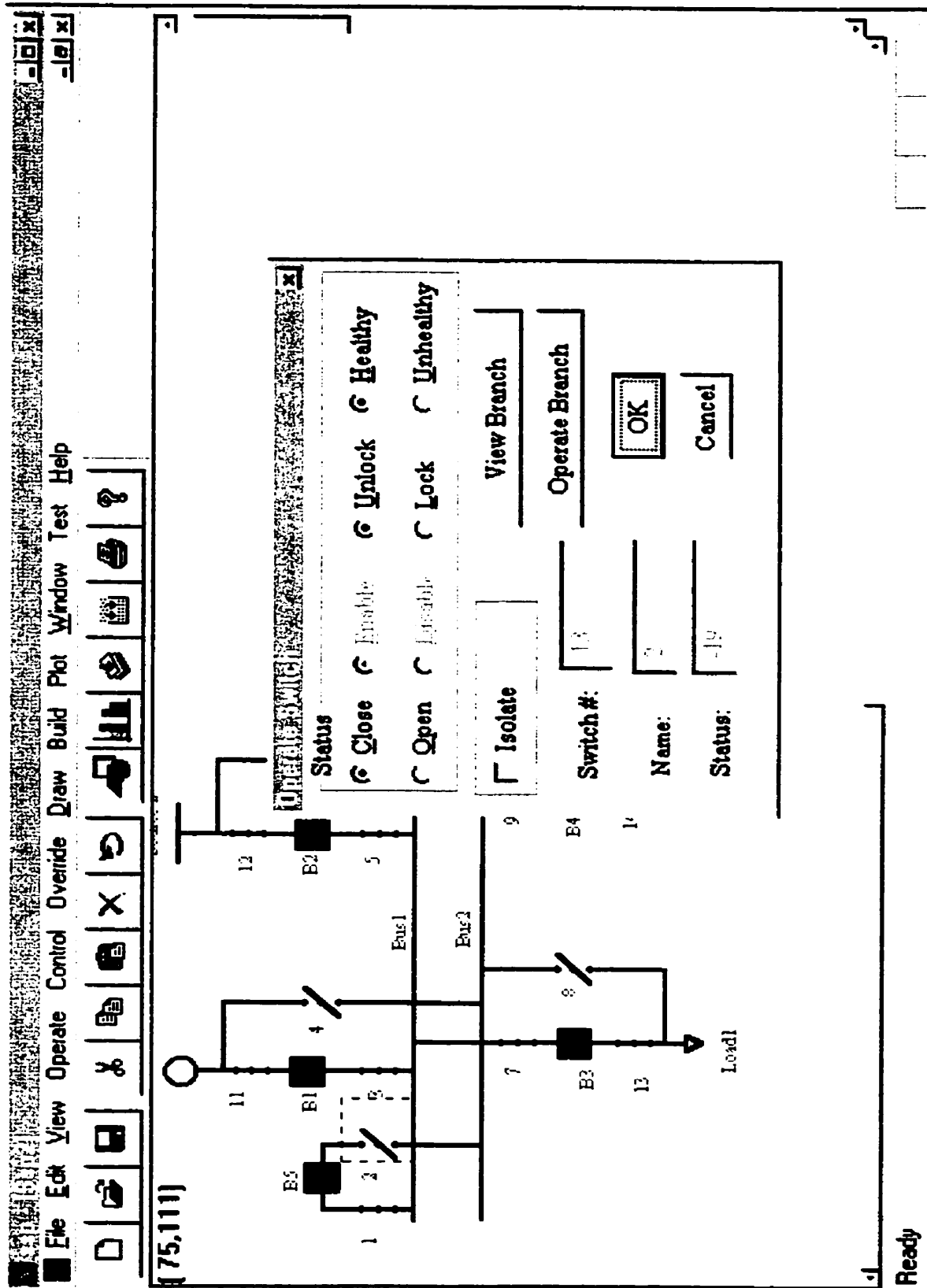
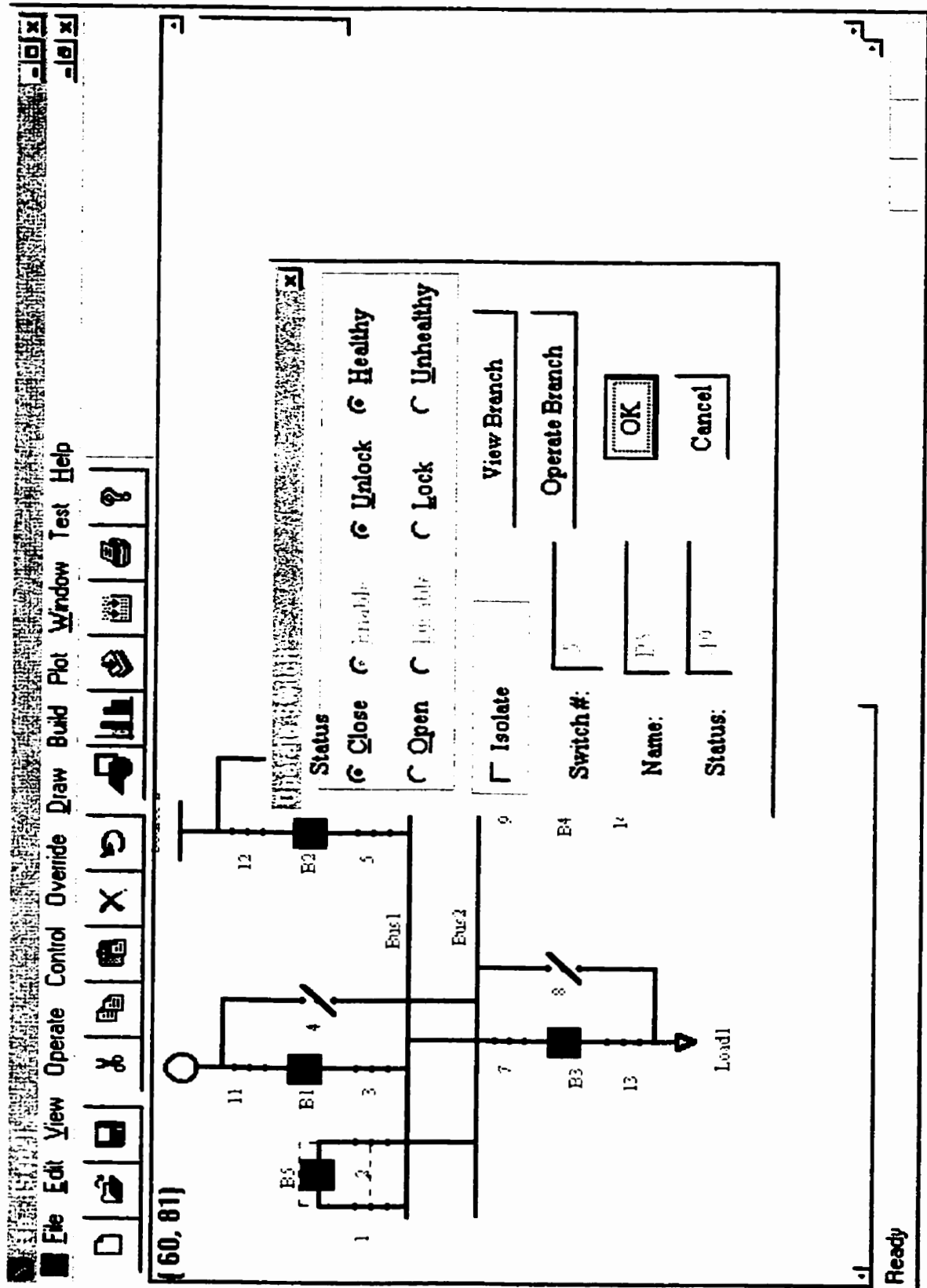
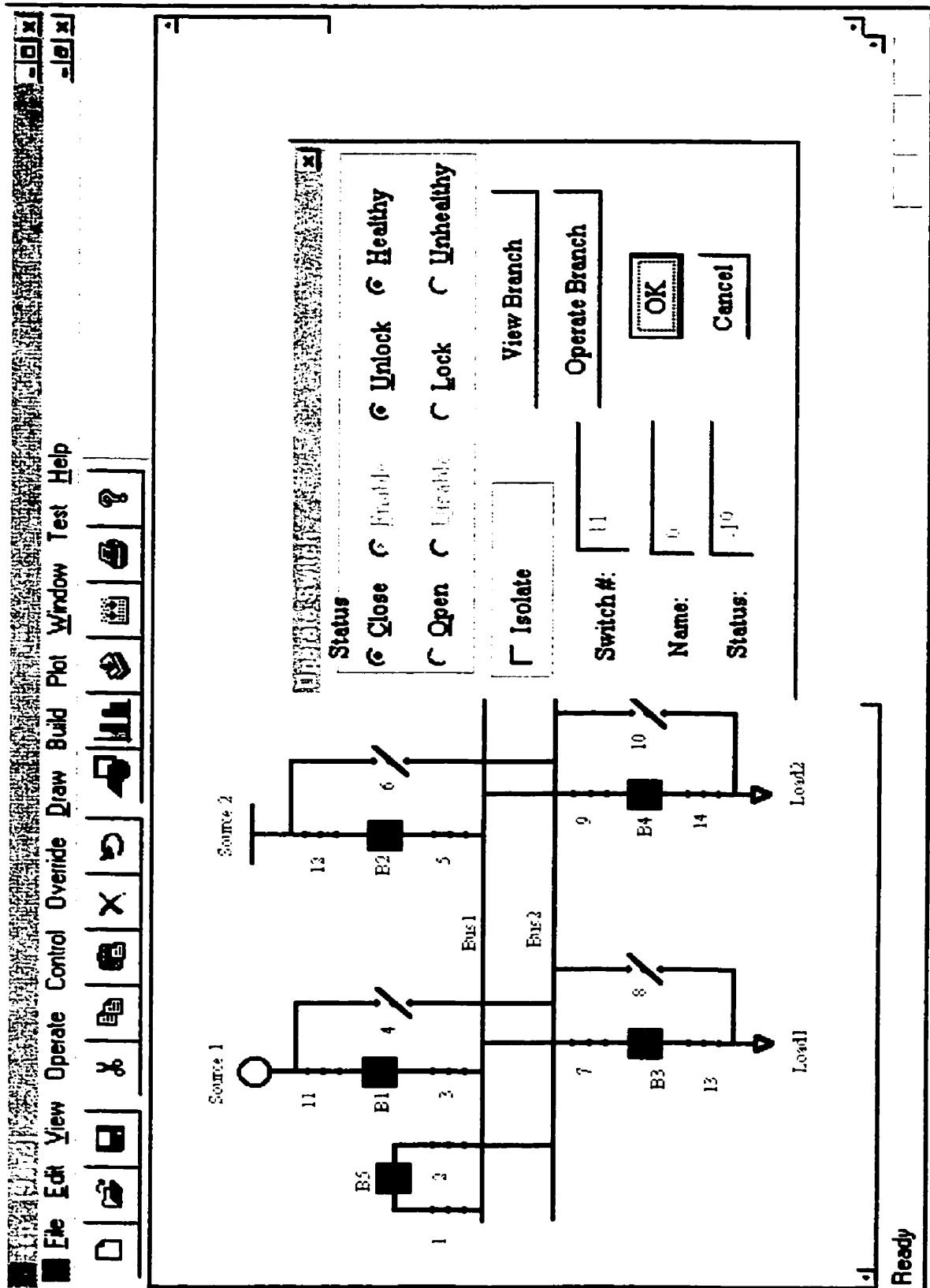


Figure H.18. The SSP has closed isolator 1 and a new command to close isolator 2 is being issued.

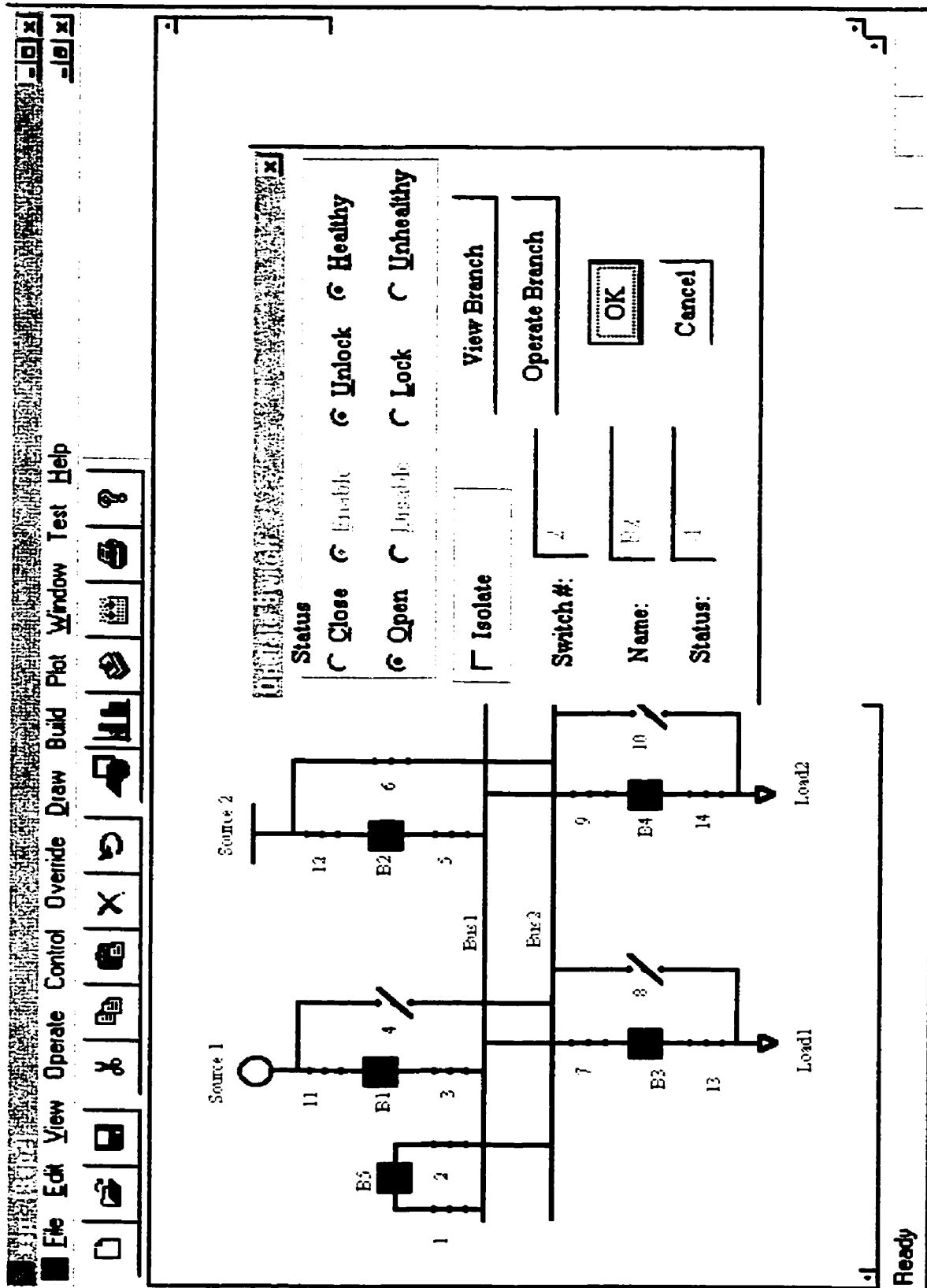


**Figure H.19.** The SSP has closed isolator 2 and a command is being issued to close circuit breaker B5.

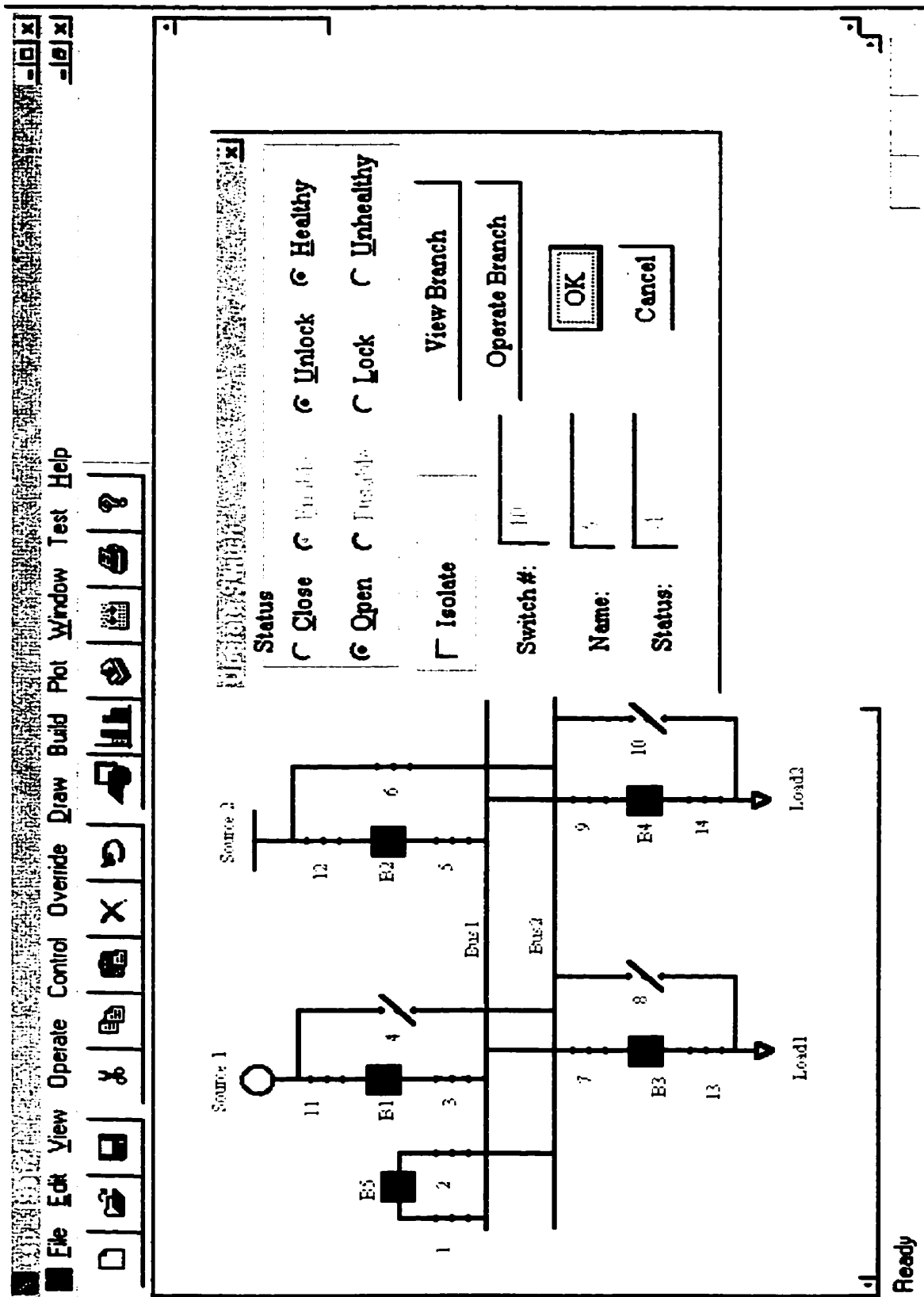




**Figure H.20.** The SSP has closed circuit breaker B5 and a command is being issued to close isolator 6.



**Figure H.21.** The SSP has closed isolator 6 and a command is being issued to open circuit breaker B2.



**Figure H.22.** The SSP has opened circuit breaker B2 and a command is being issued to open isolator 5.

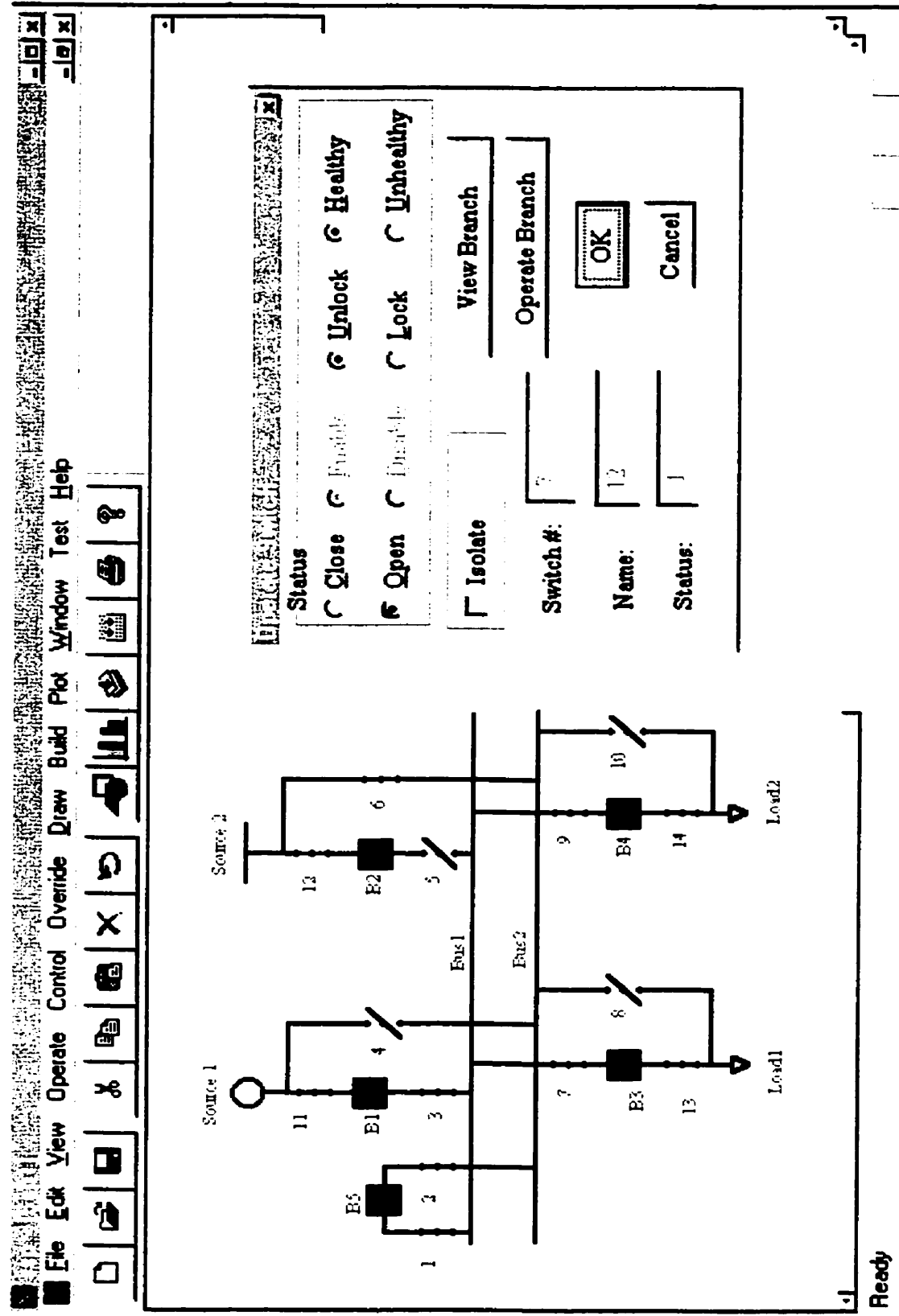


Figure H.23. Command being issued to open isolator 12.

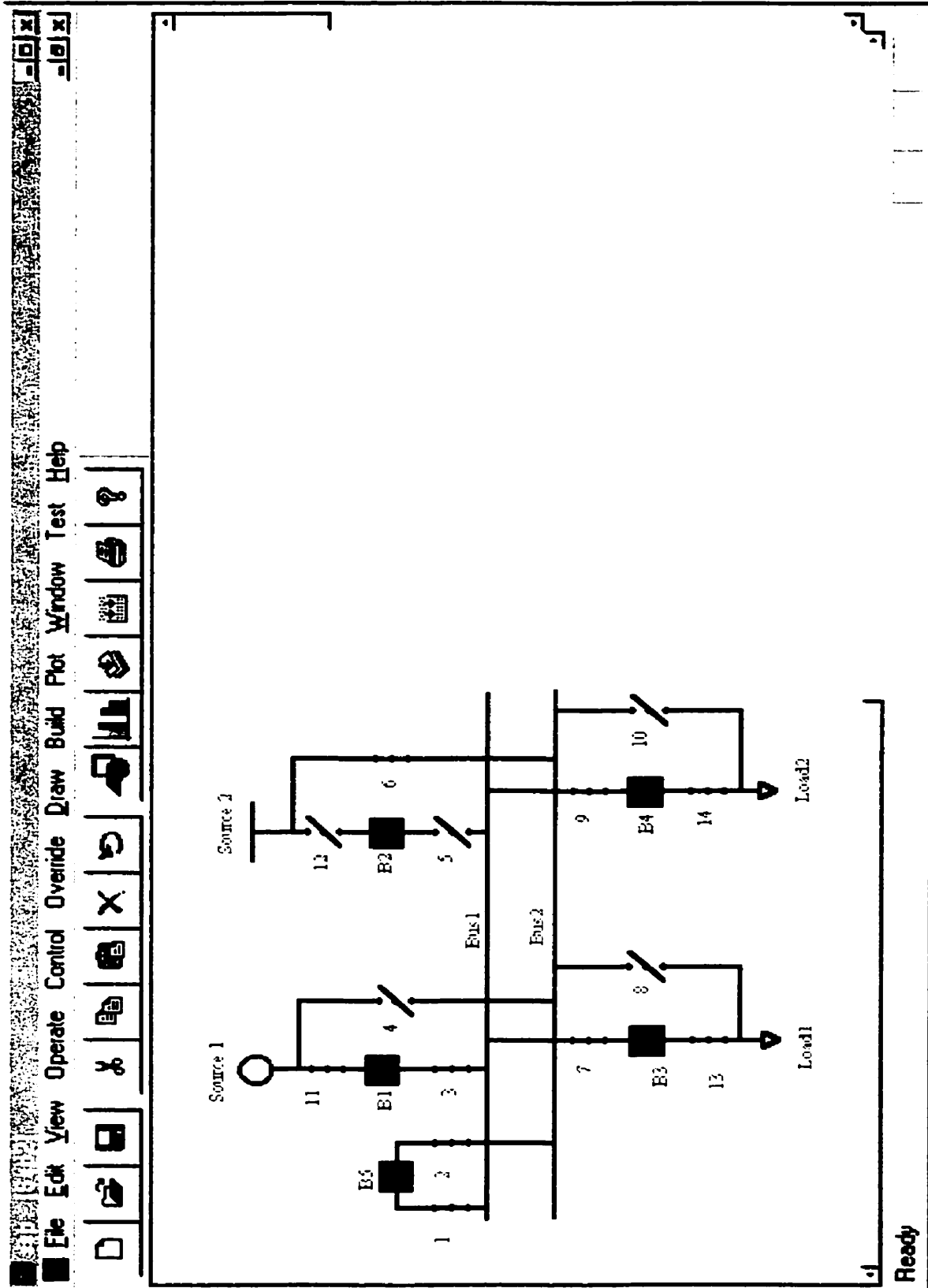


Figure H.24. The SSP opens isolator 12.

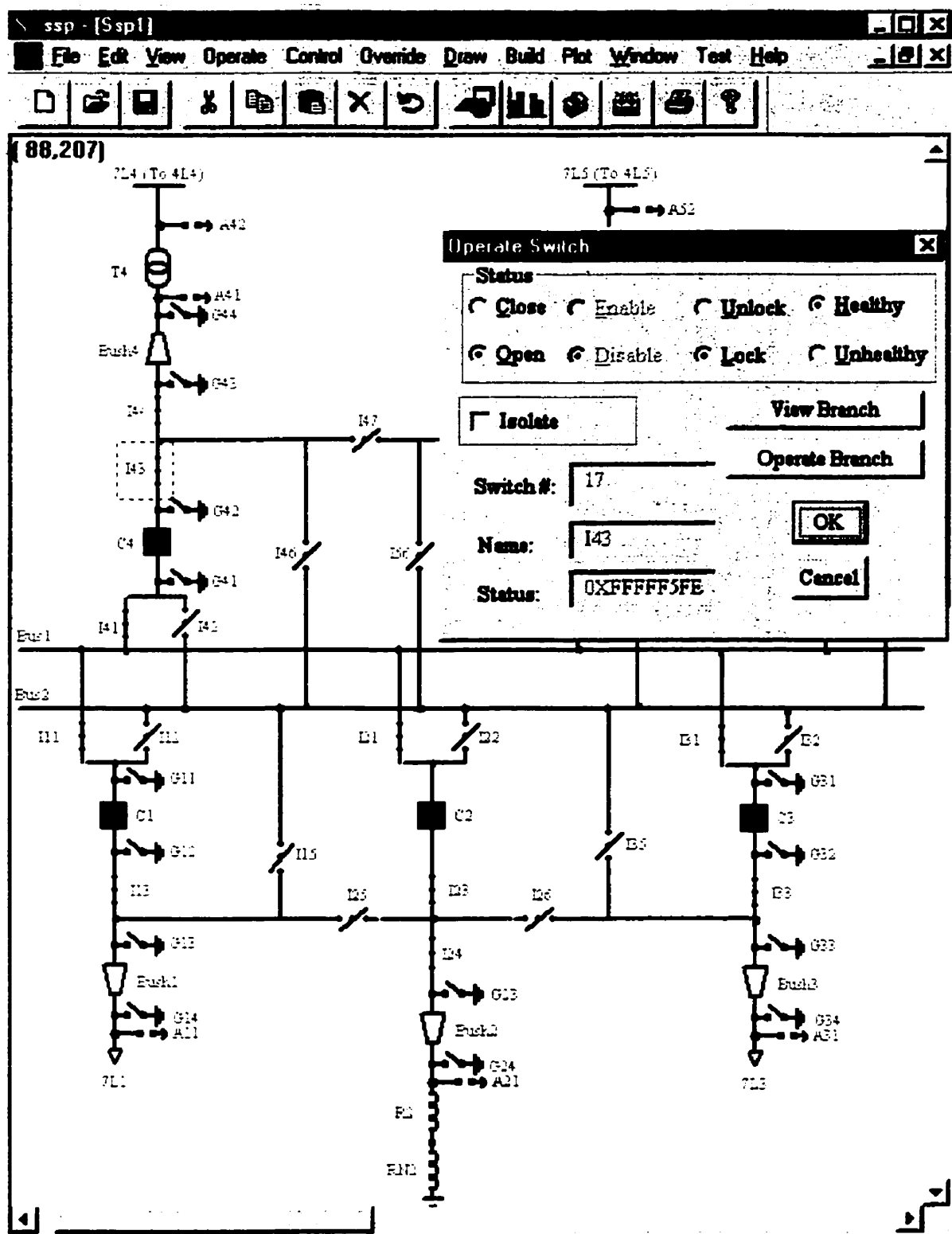
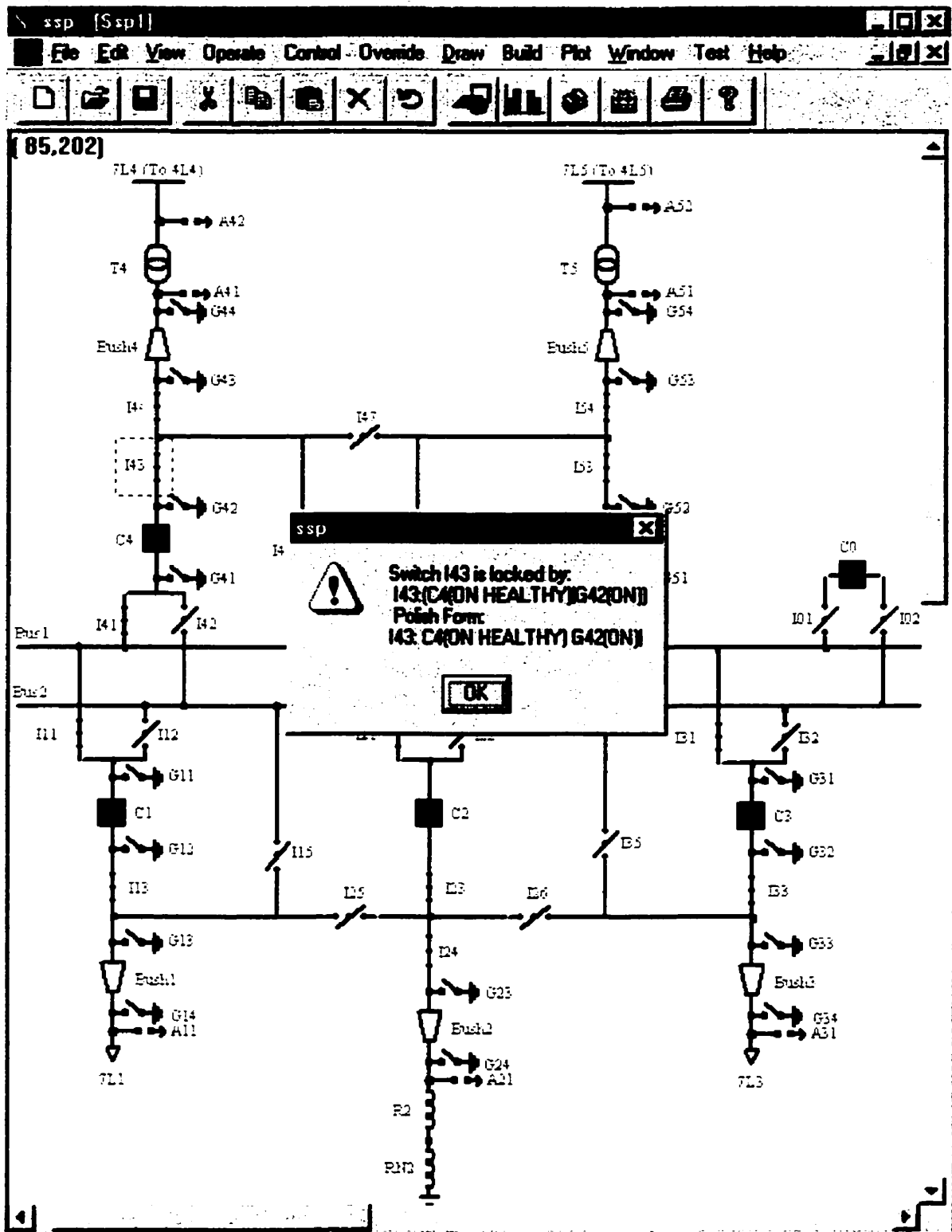


Figure H.25. Command being issued to open isolator I43.



**Figure H.26.** The SSP refuses to operate I43 and displays the reason for the refusal.

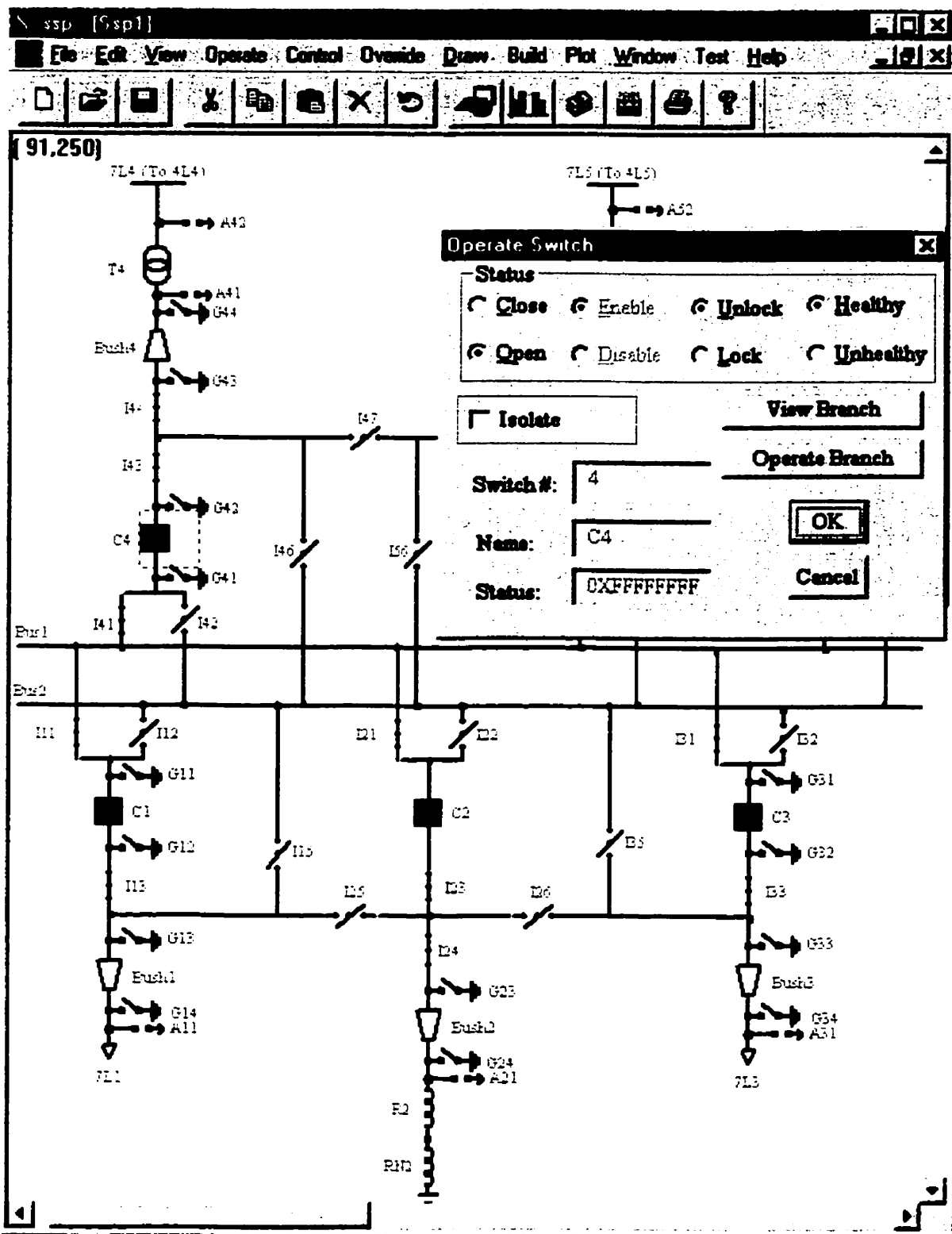
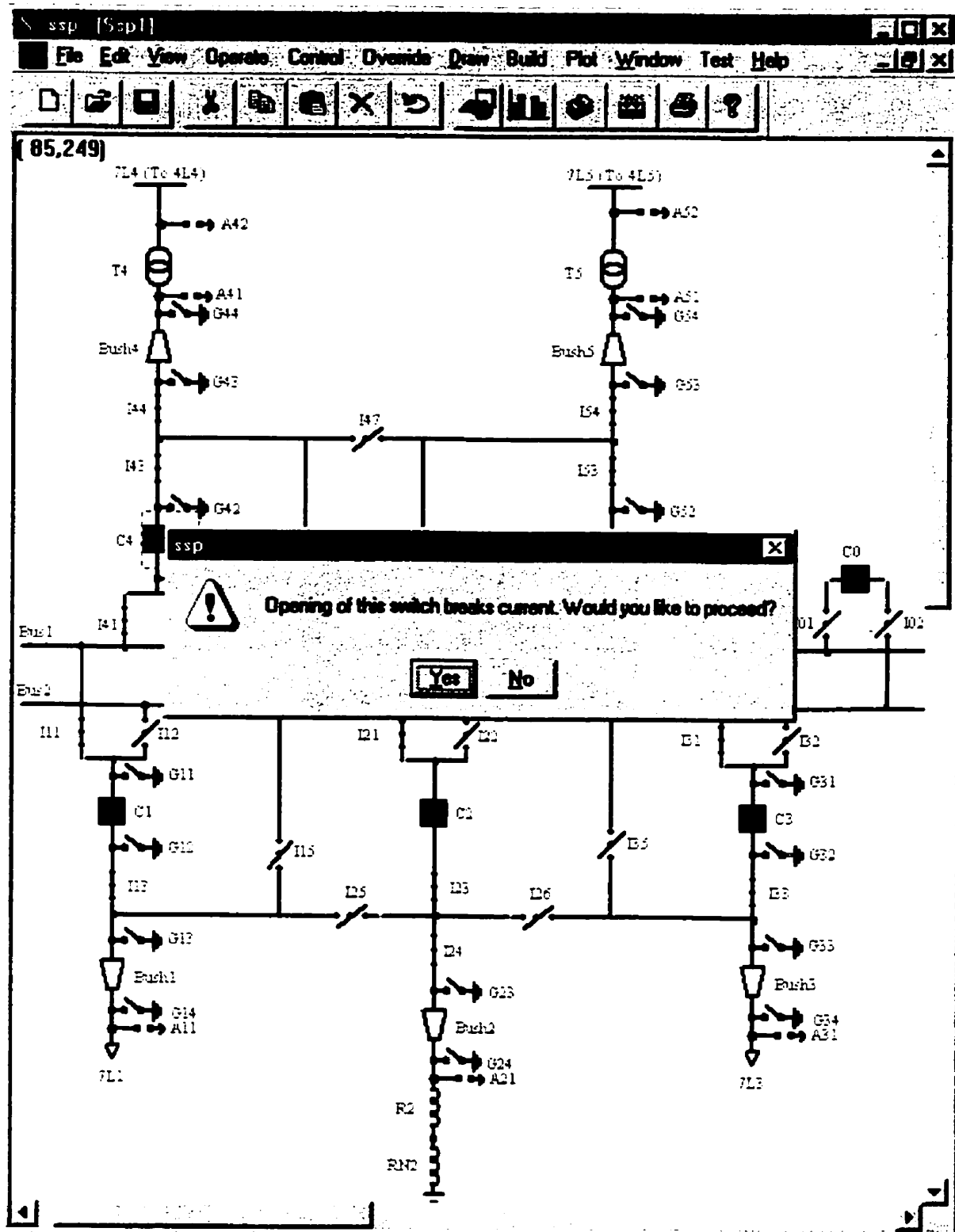
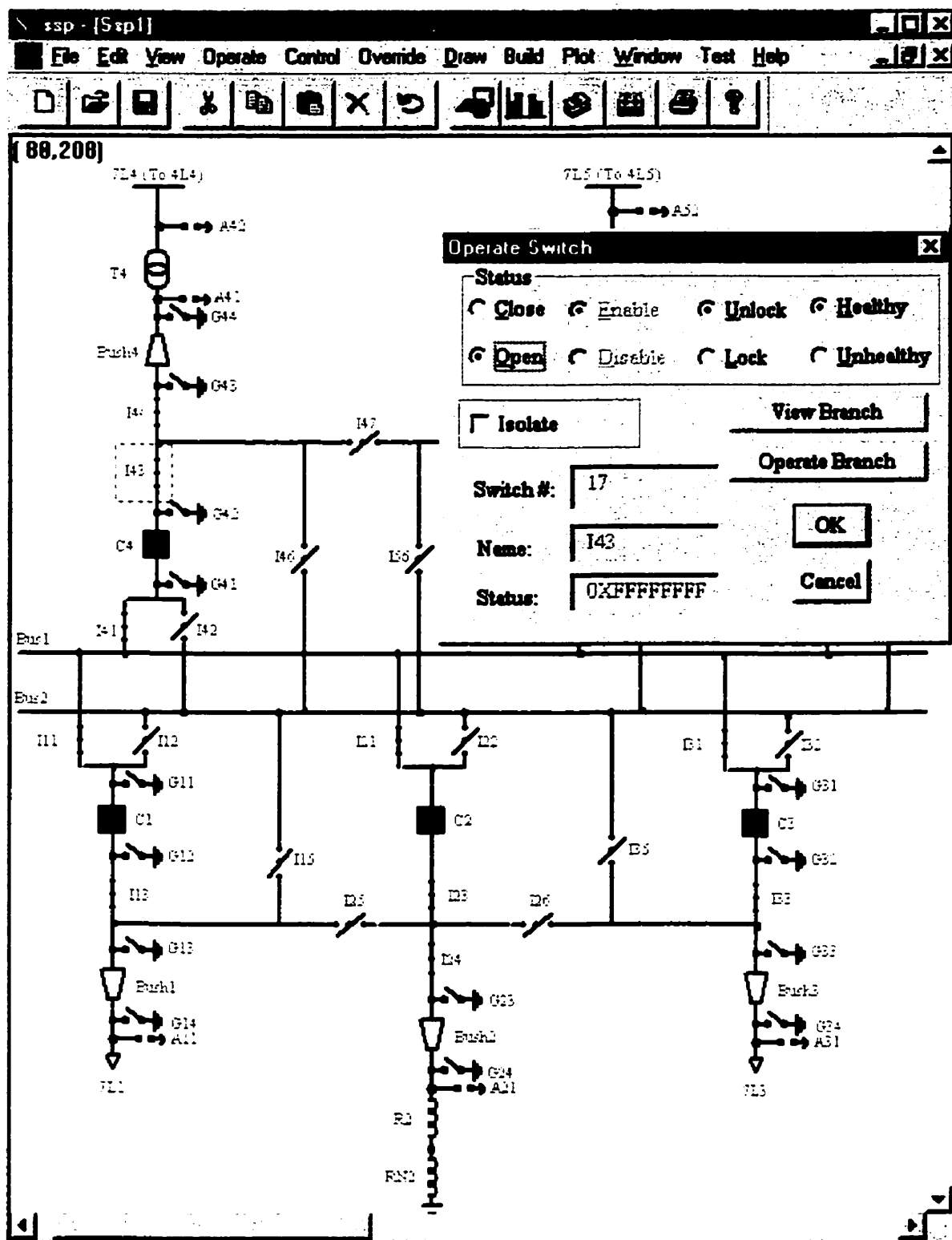


Figure H.27. Command being issued to open circuit breaker C4.

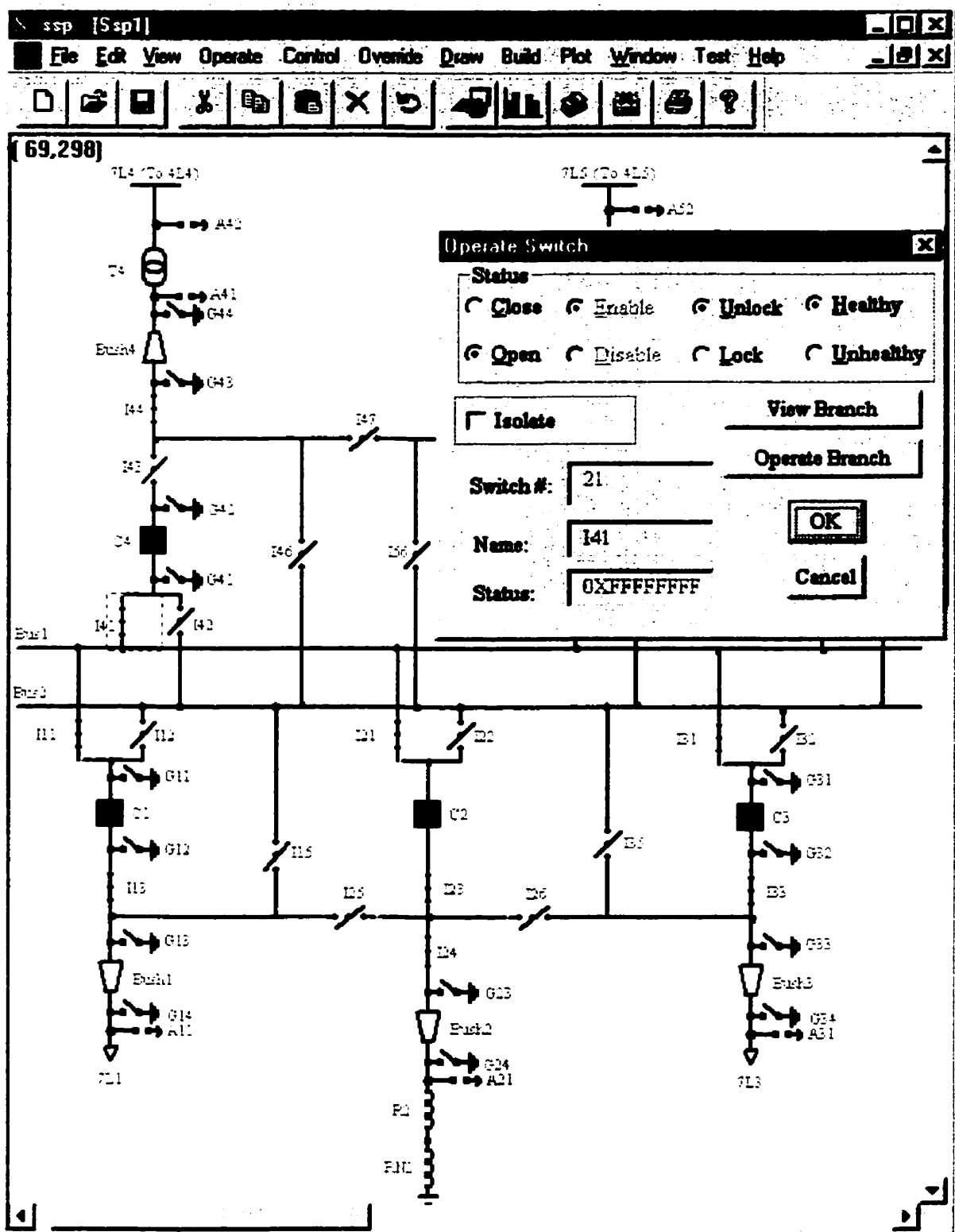




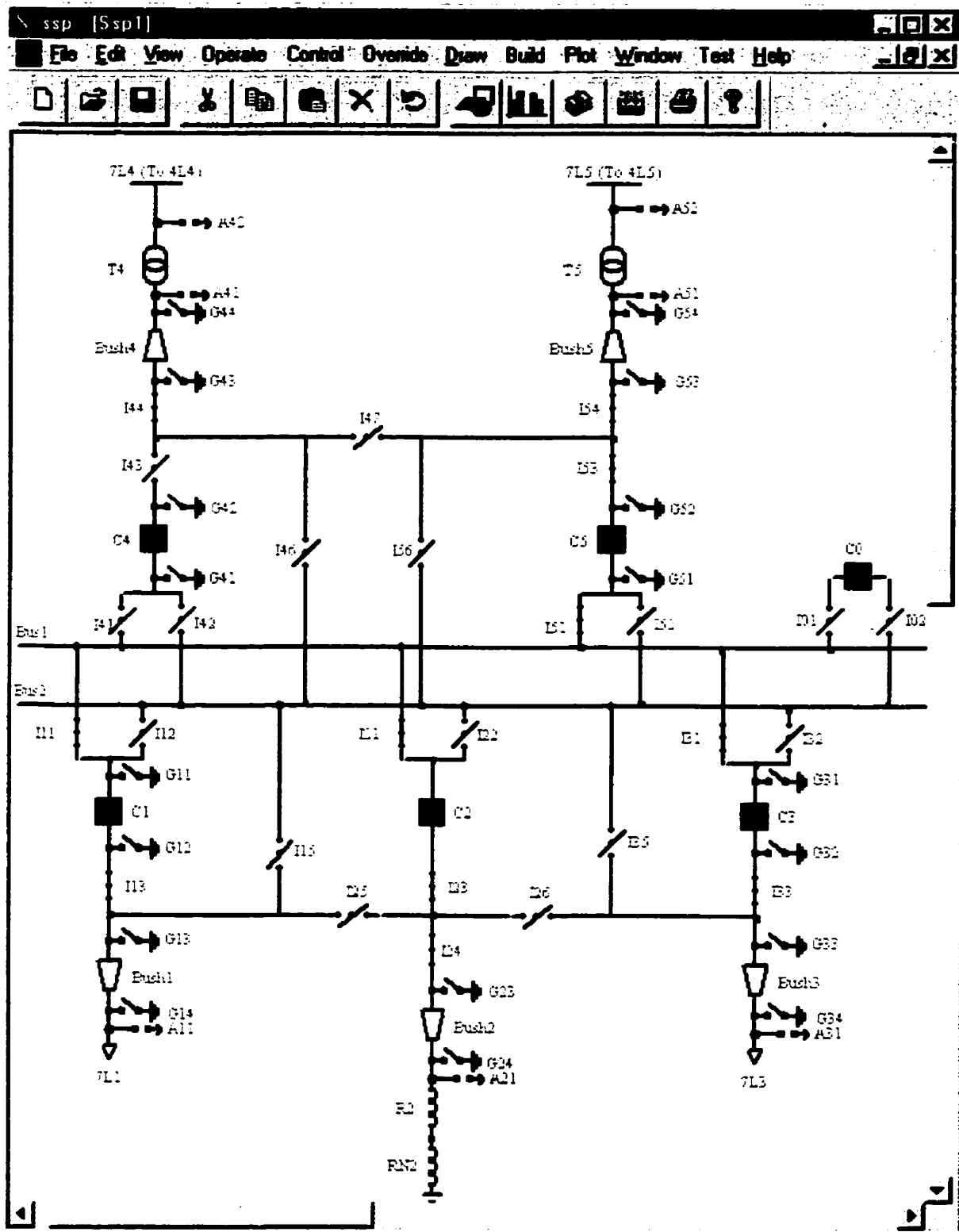
**Figure H.28.** The SSP warns about the interruption of current, and gives a chance to cancel the closing operation.



**Figure H.29.** The SSP has closed C4 and a command is being issued to open isolator I43.



**Figure H.30.** The SSP has opened isolator I43 and a command is being issued to open isolator I41.



**Figure H.31.** The SSP has opened isolator I41 and enabled grounding switches G41 and G42.

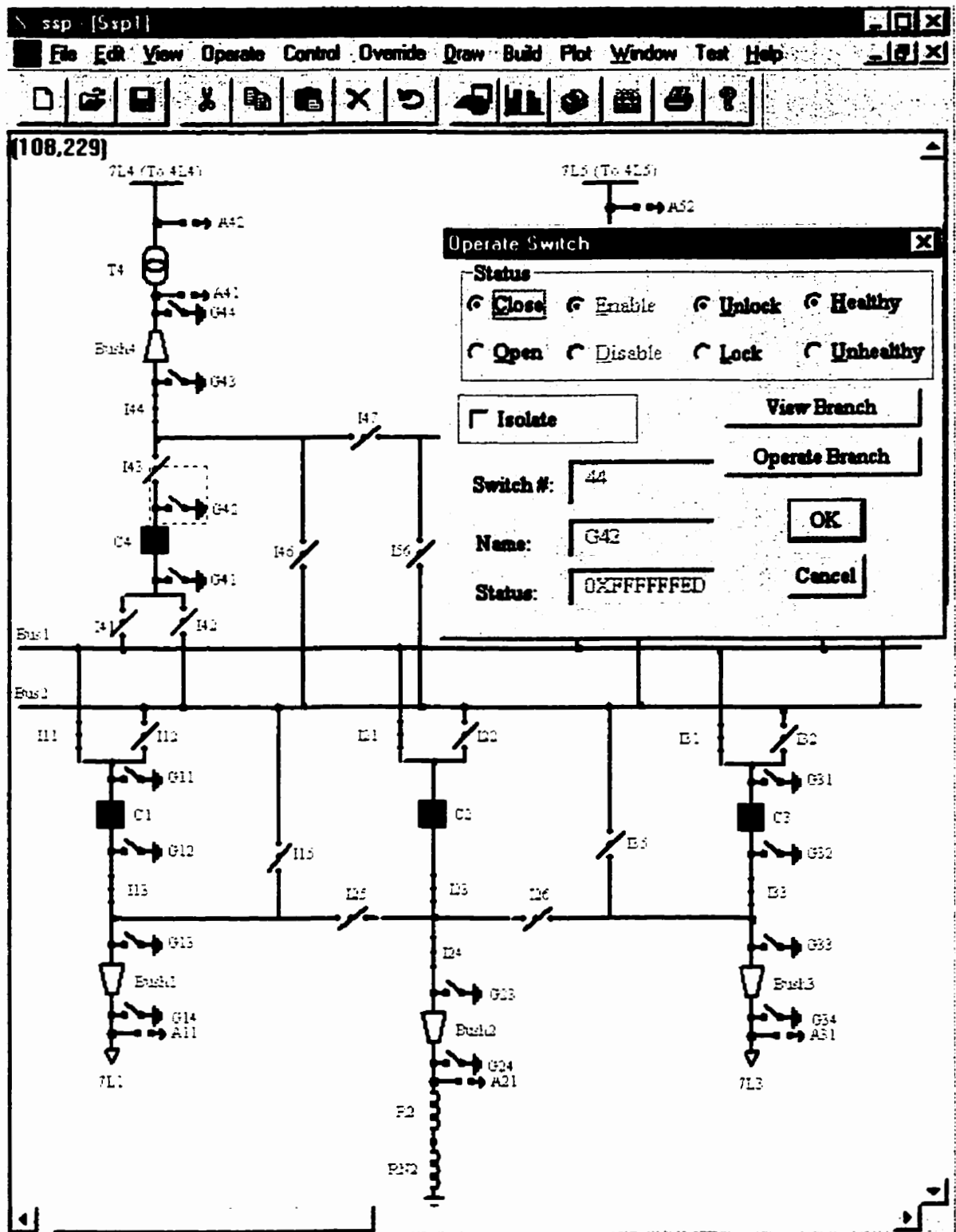
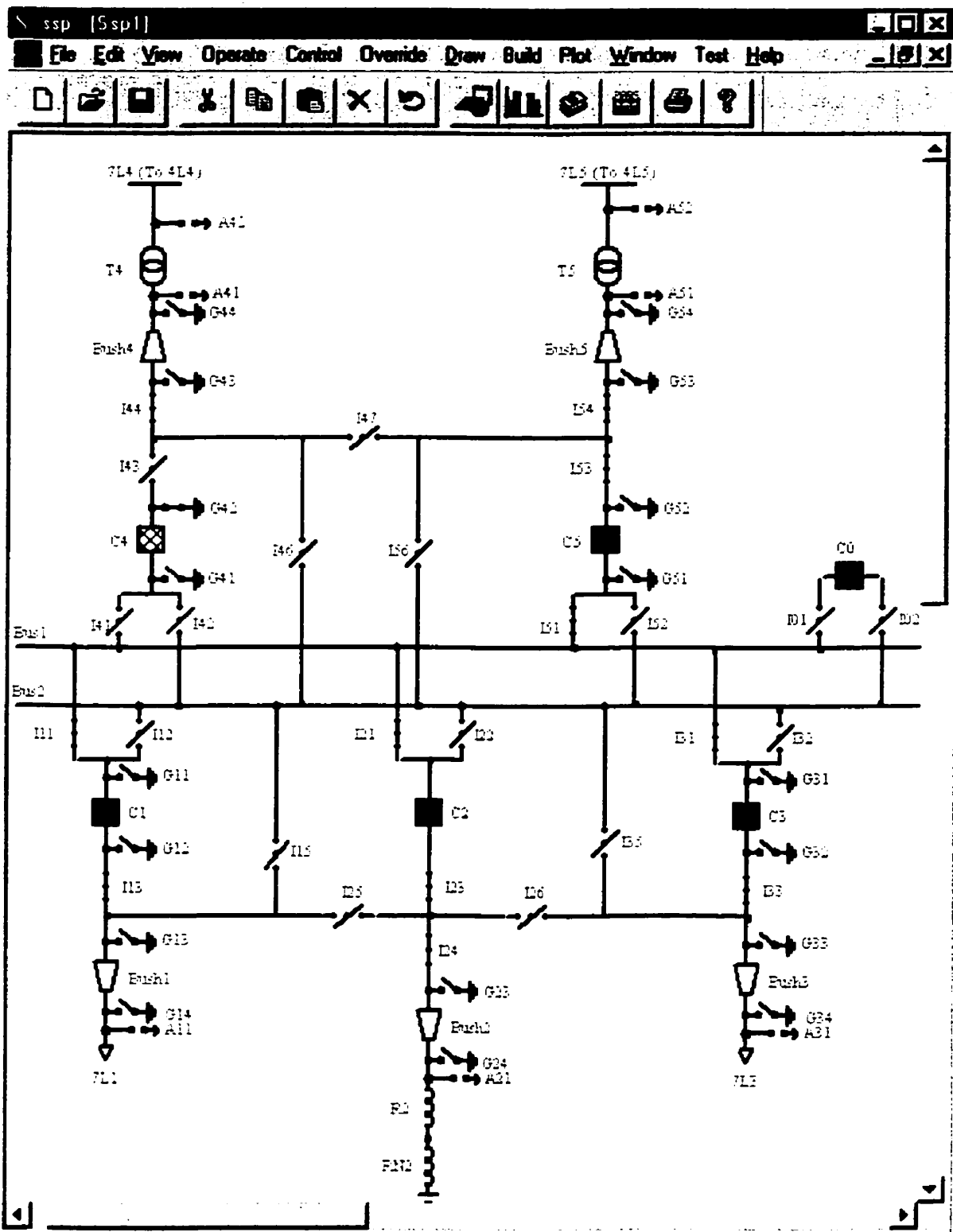


Figure H.32. Command being issued to close grounding switch G42.



**Figure H.33.** The SSP closes grounding switch G42 and disables isolator I43 and circuit breaker C4.

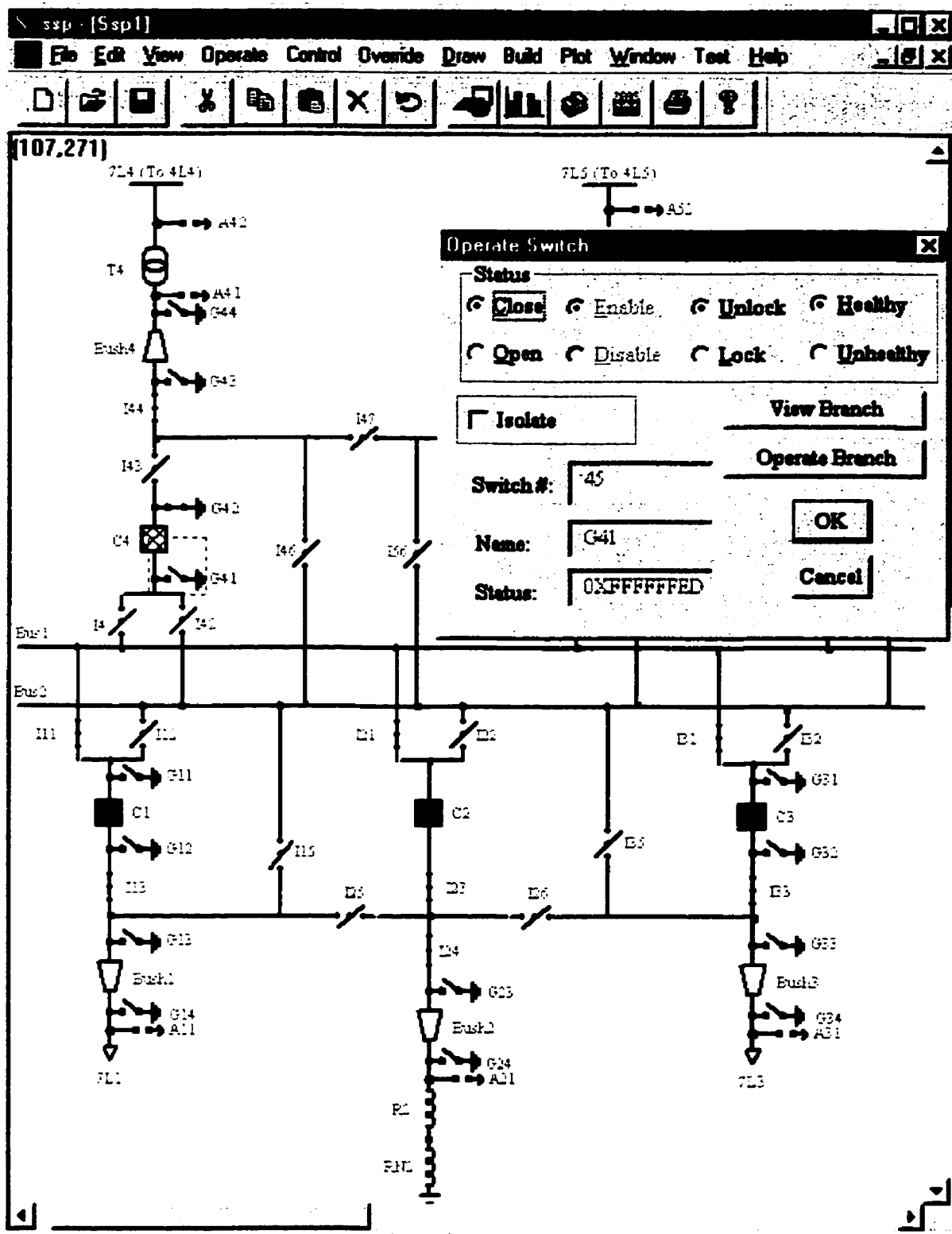
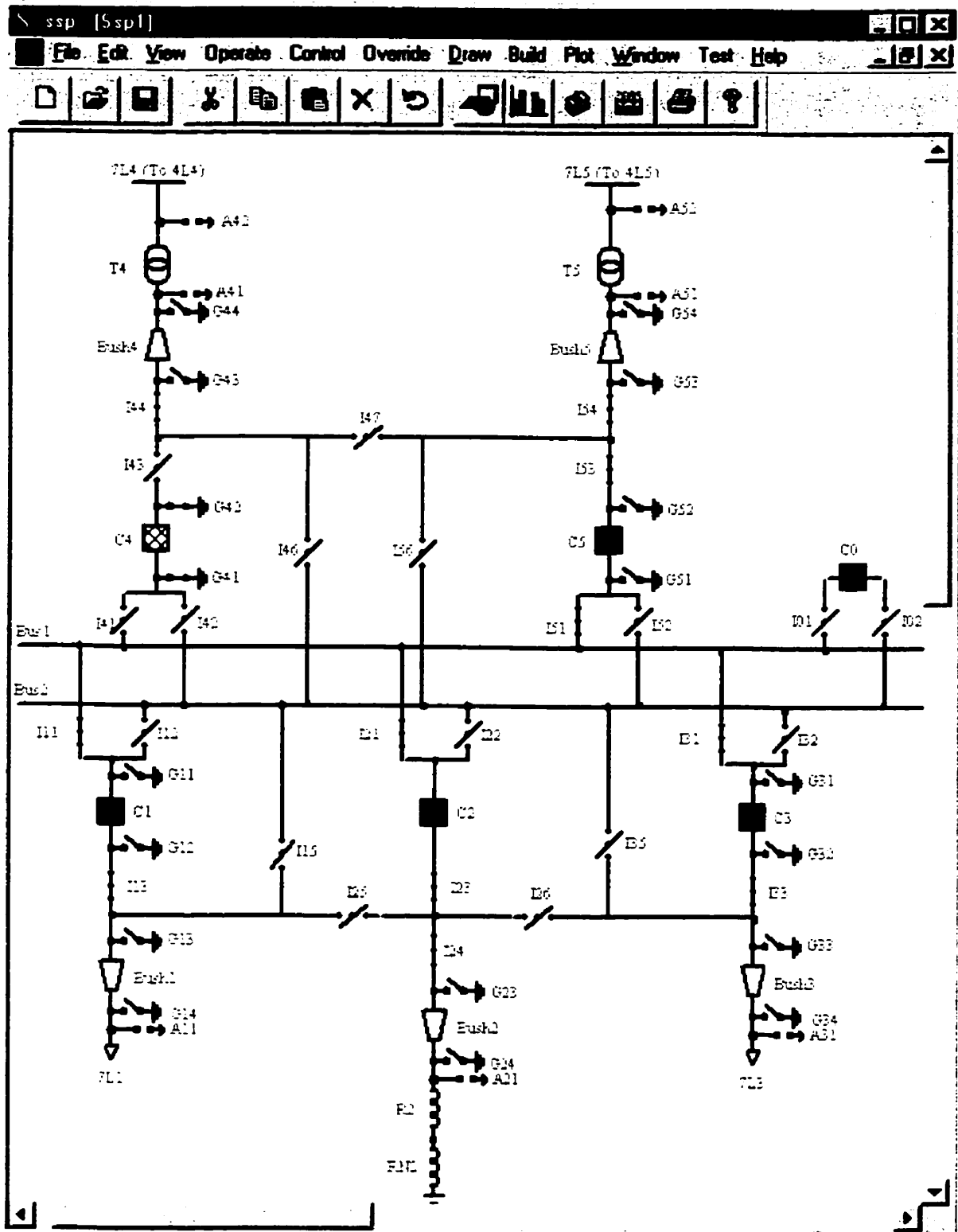
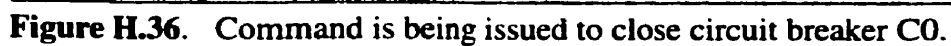


Figure H.34. Command being issued to close grounding switch G41.



**Figure H.35.** The SSP has closed grounding switch G41 and has disabled isolators I41 and I42.





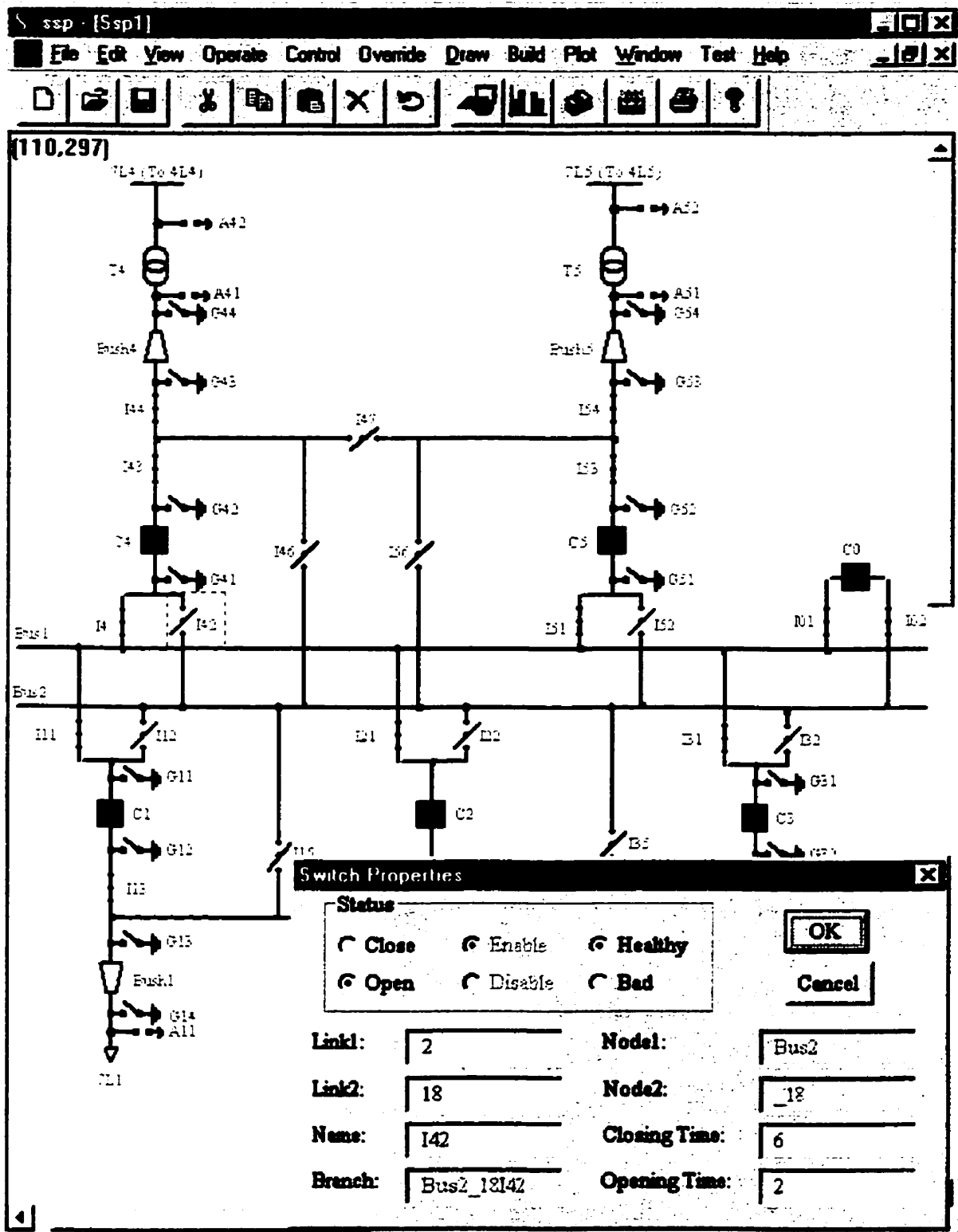


Figure H.37. Switches I12, I22, I32, I42 and I52 are enabled.

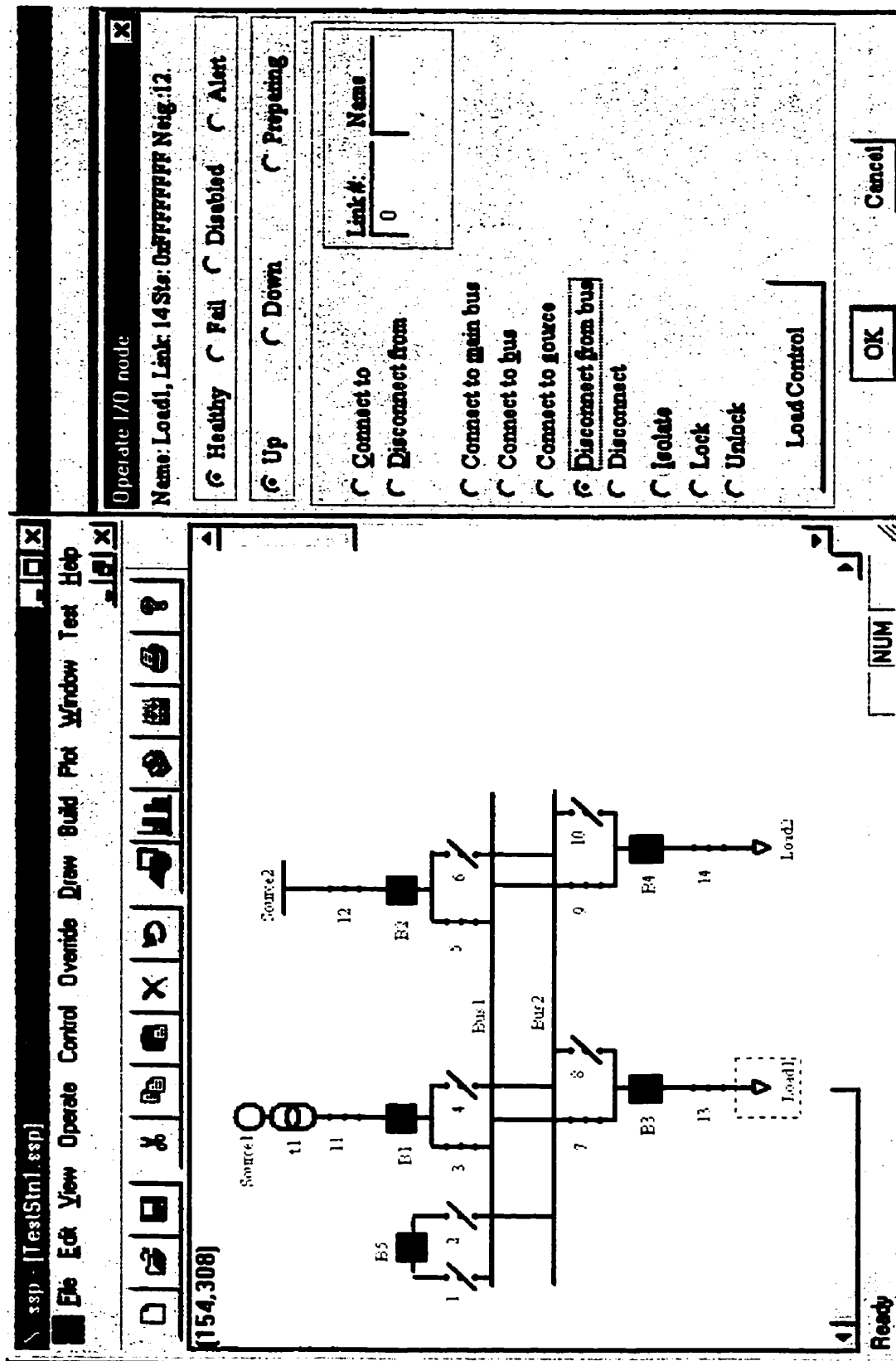


Figure H.38. Command being issued to disconnect Load1 from the bus.

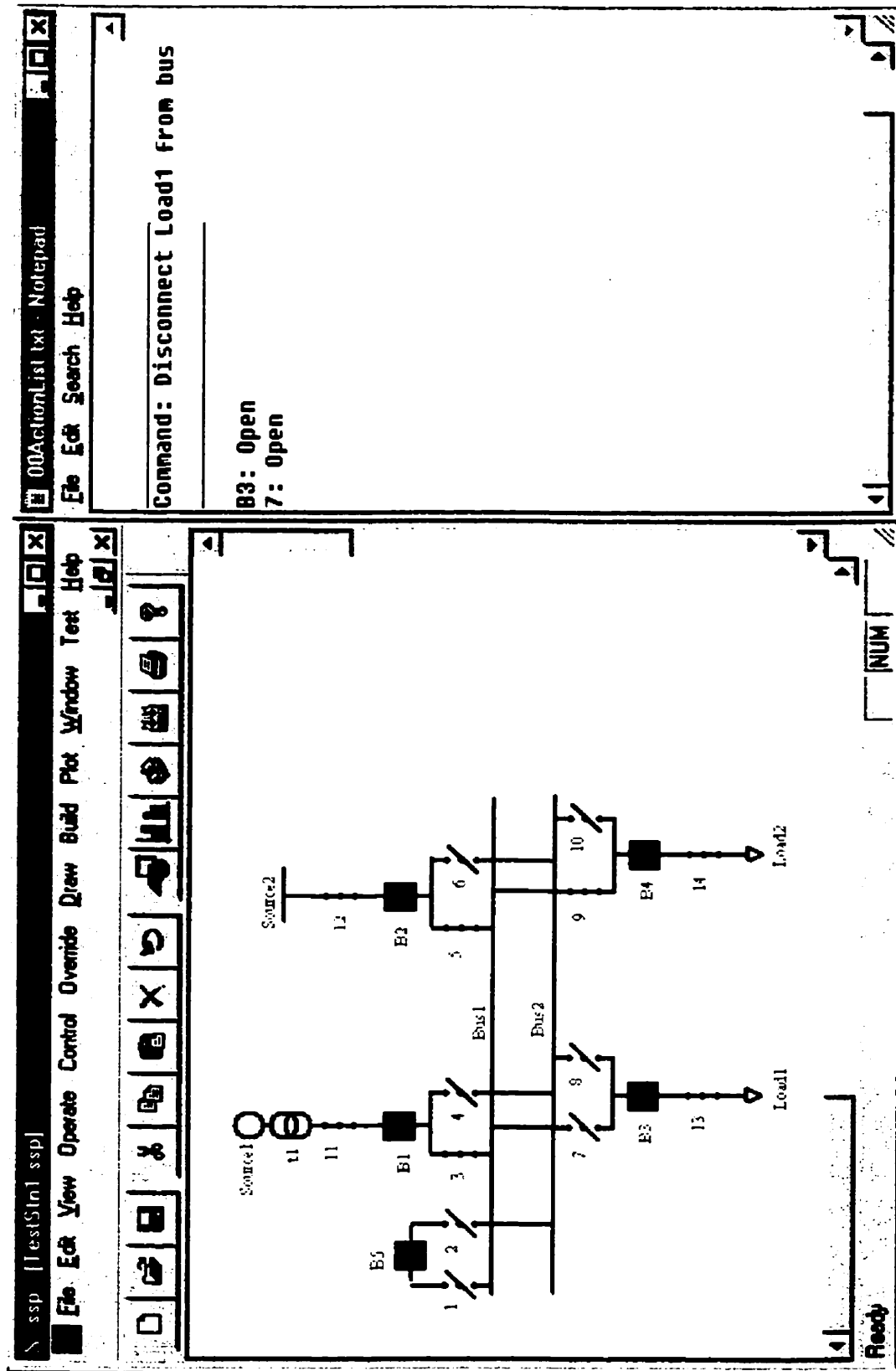


Figure H.39. The SSP disconnects Load1 from Bus1.

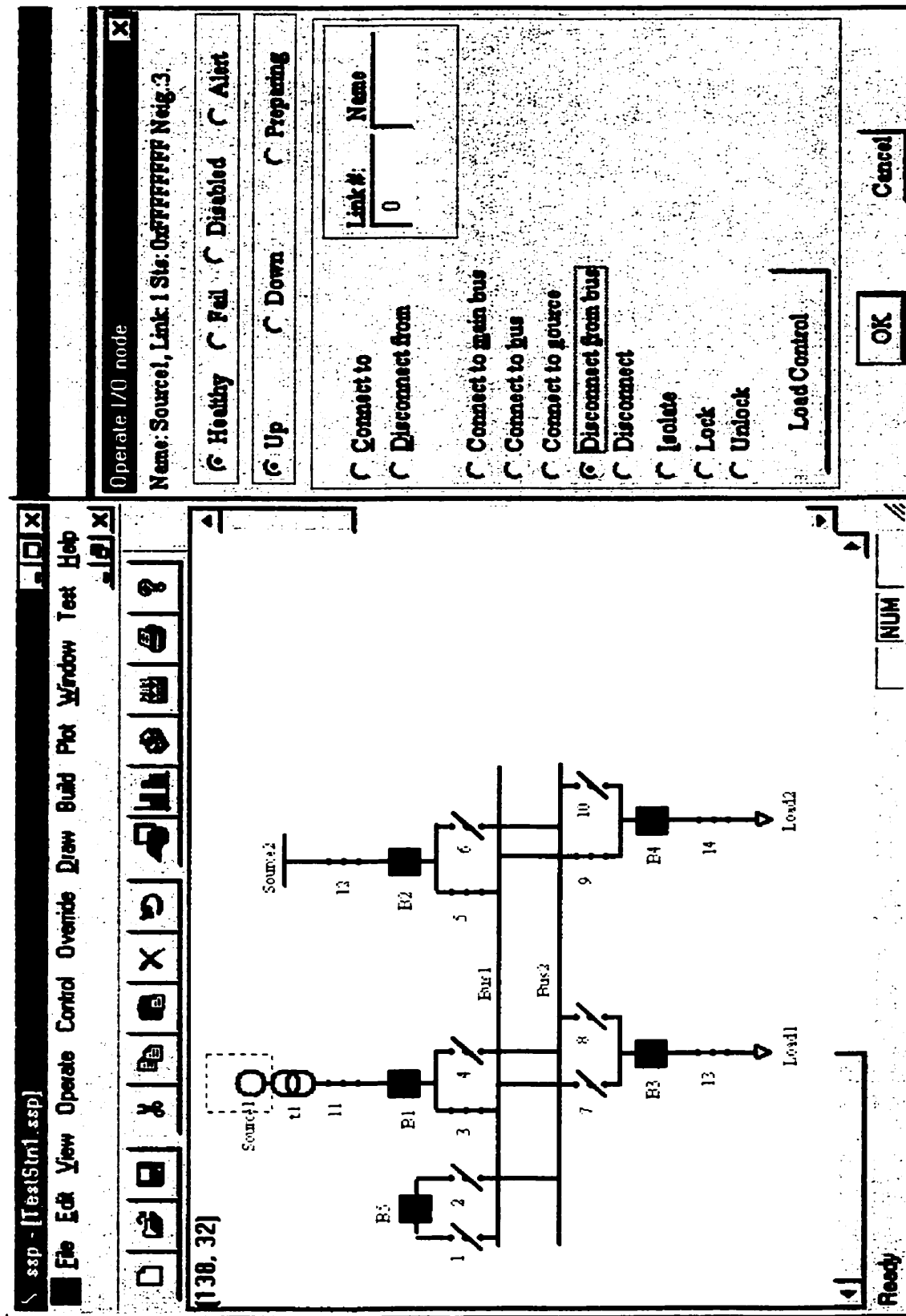


Figure H.40. Command being issued to disconnect Source1 from busbars.

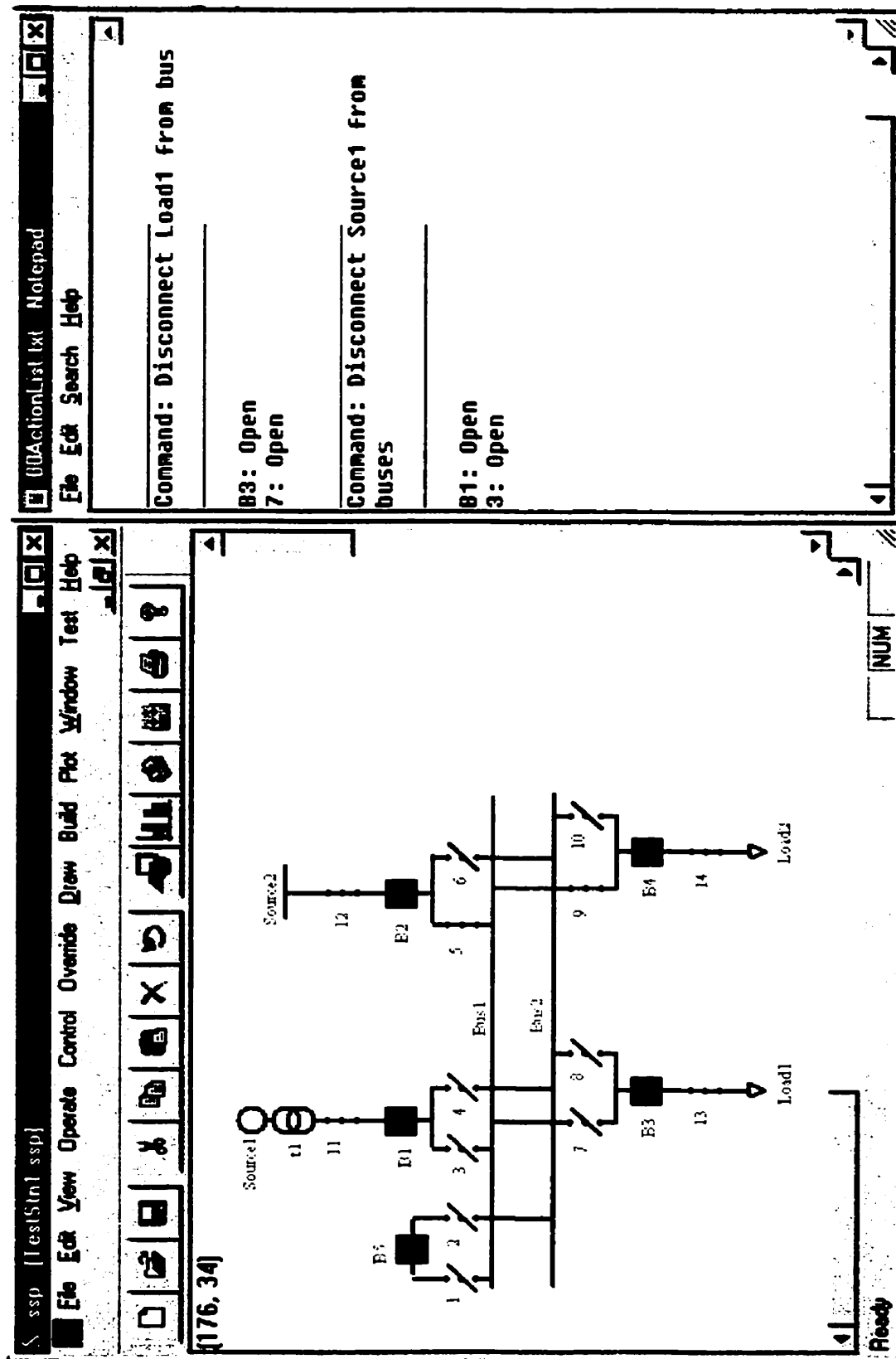


Figure H.41. The SSP disconnects Source1 from busbars.

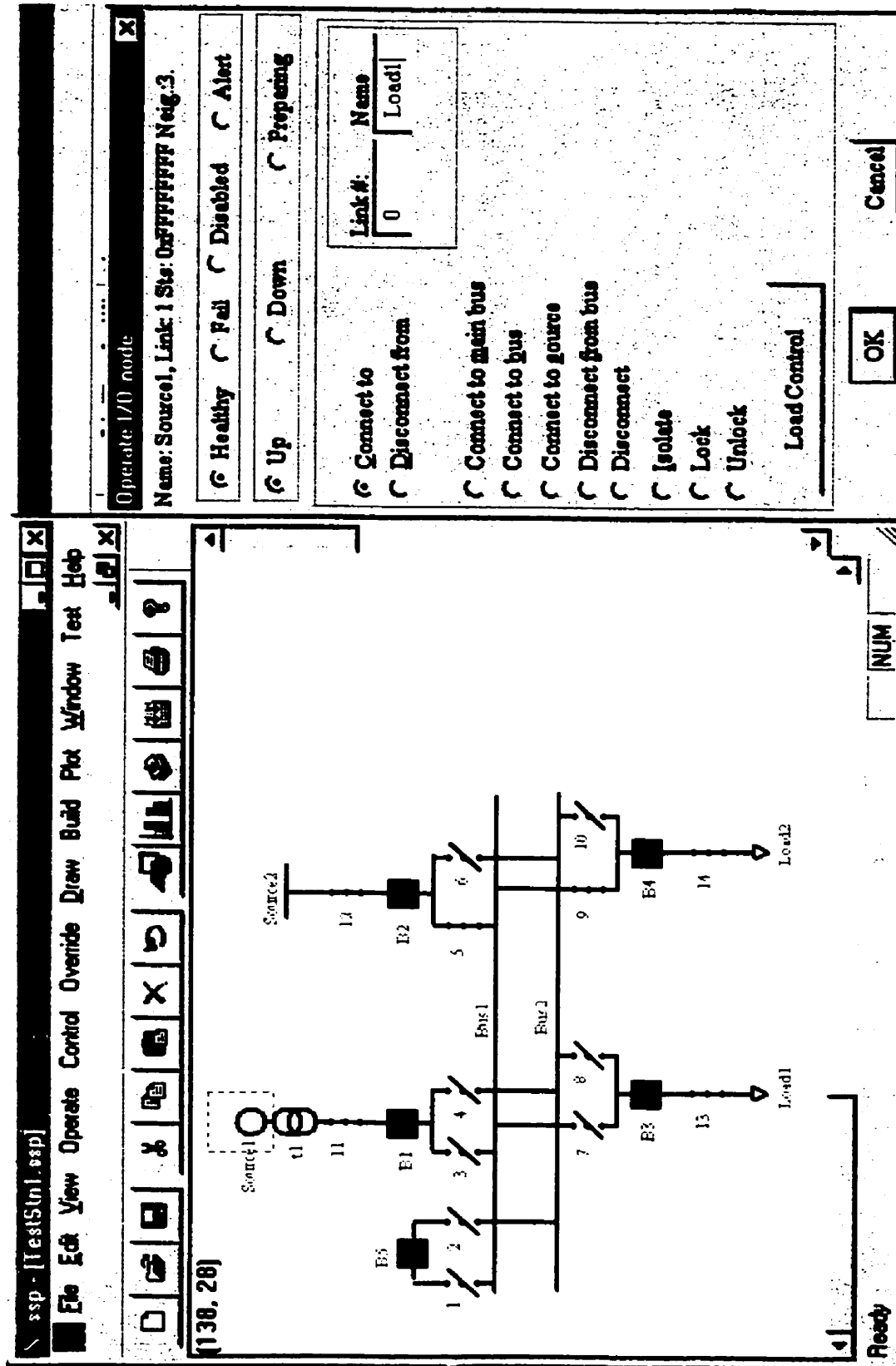


Figure H.42. Command being issued to connect Source1 to Load1.

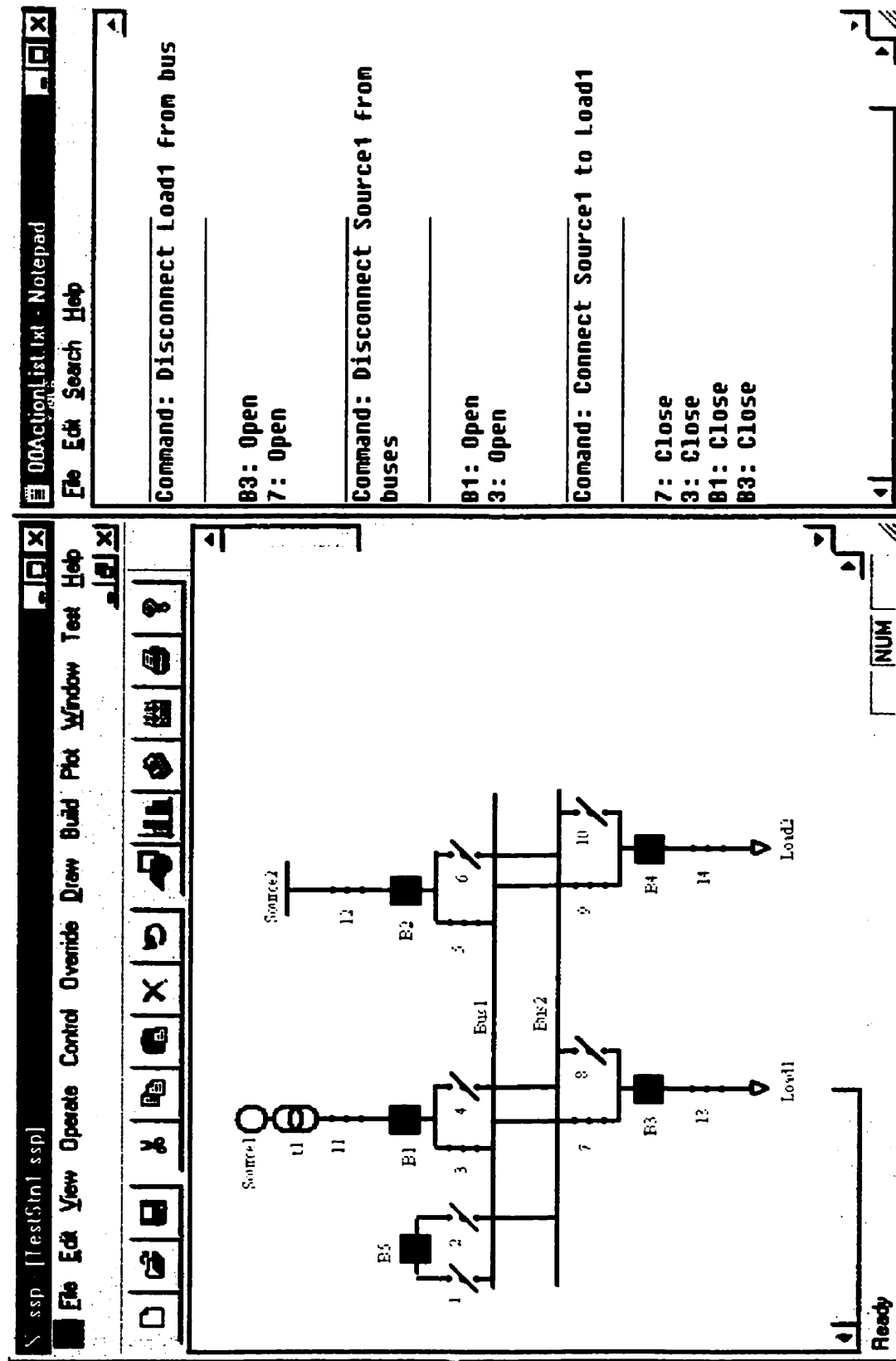


Figure H.43. The SSP connects Source1 with Load1.



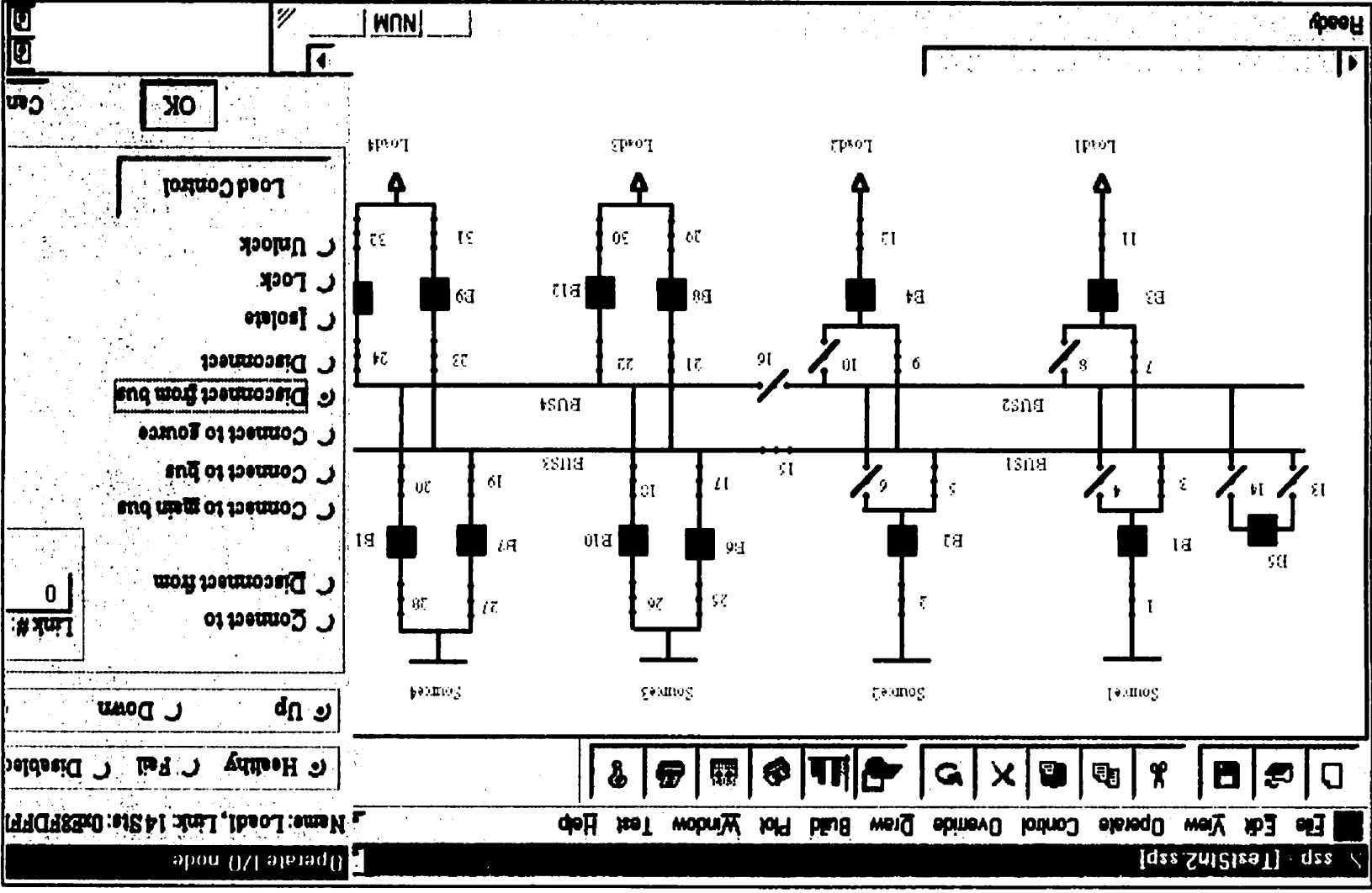


Figure H.44. Command to disconnect Load1 from the bus.

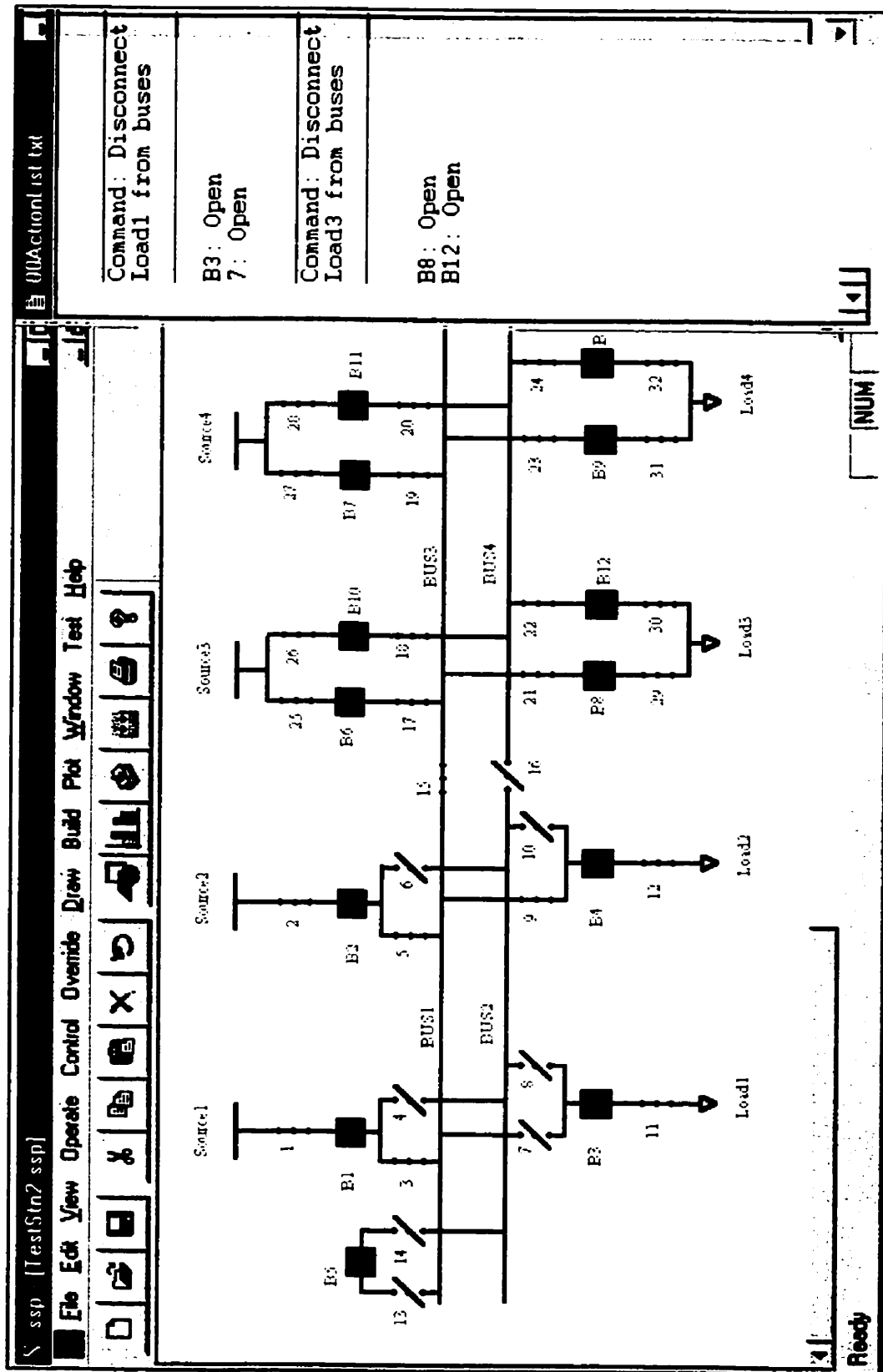


Figure H.45. State of the substation after disconnecting Load1 and Load3 from the busbars.

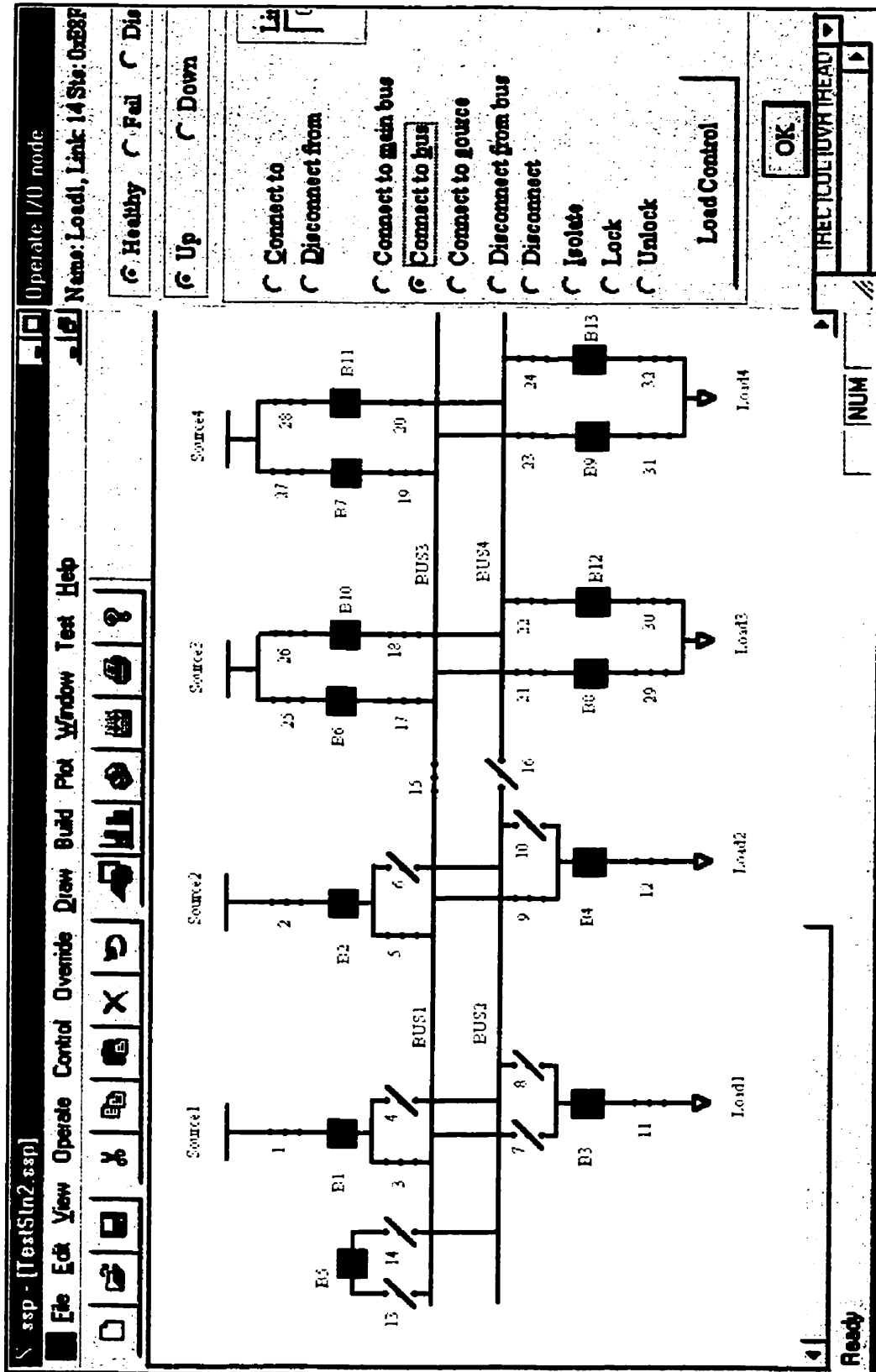


Figure H.46. Command to connect Load1 to the busbar.

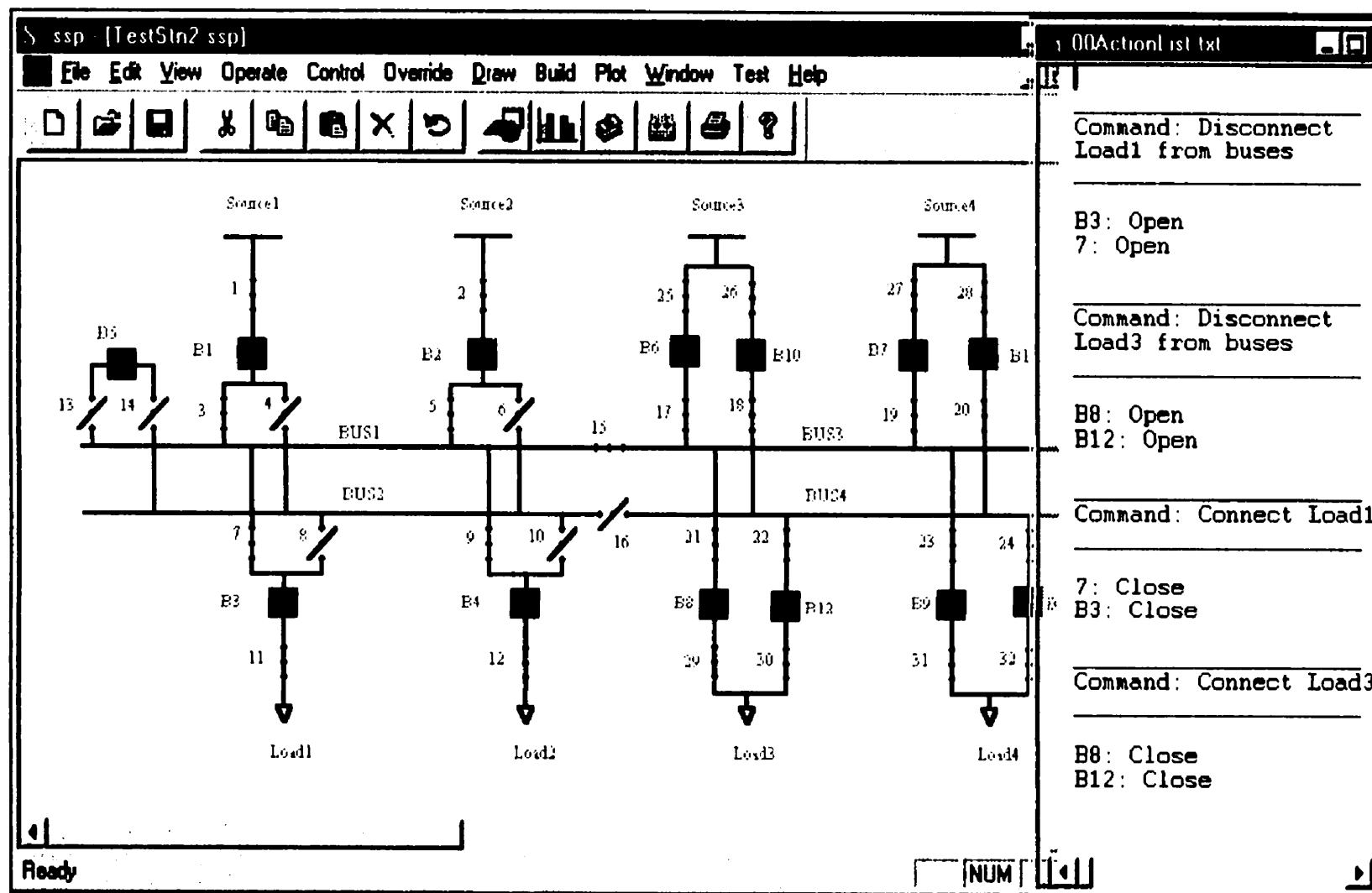


Figure H.47. Load1 and Load3 connect back to the busbars.

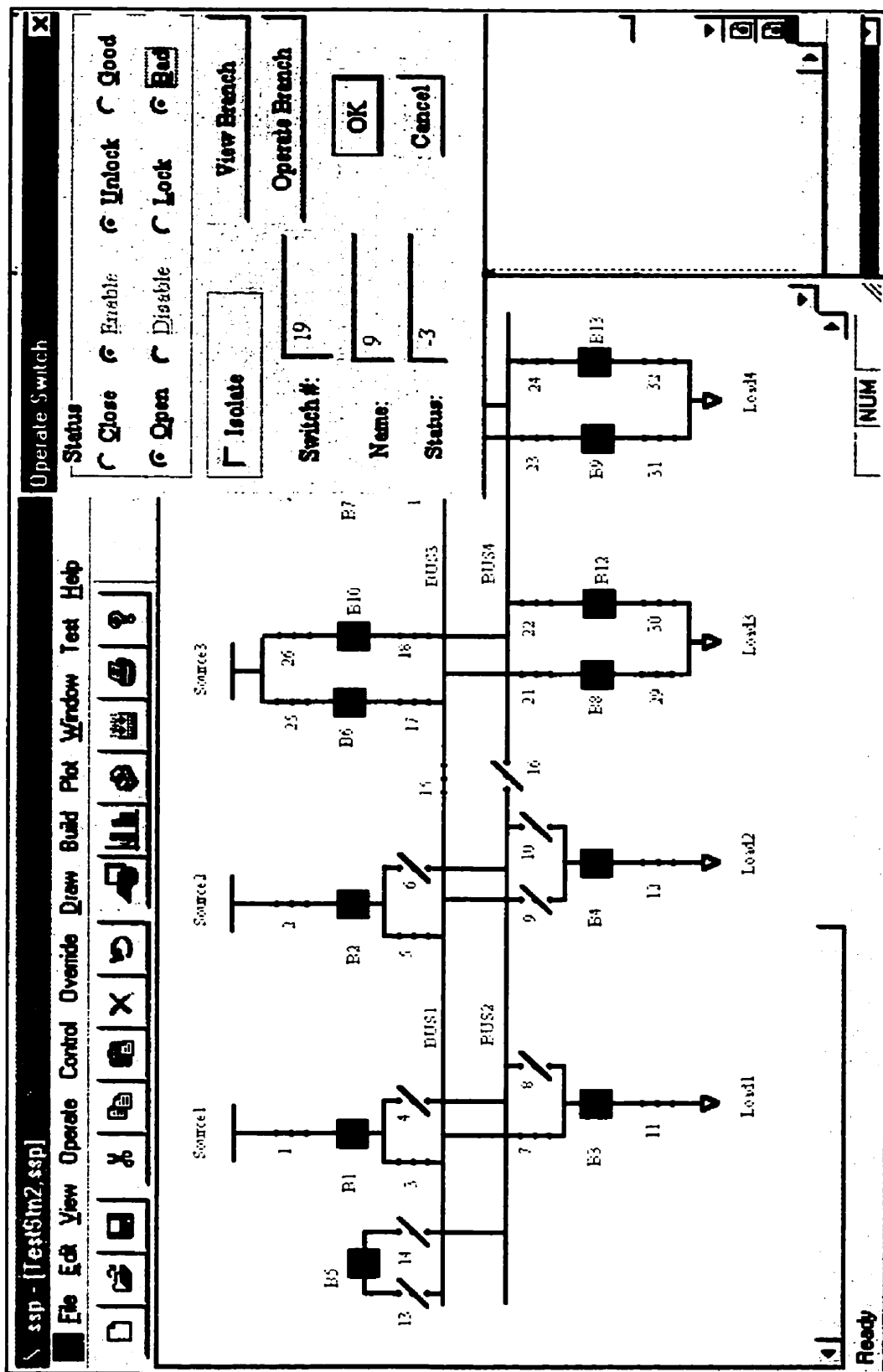


Figure H.48. Declaring isolator 9 as unhealthy.

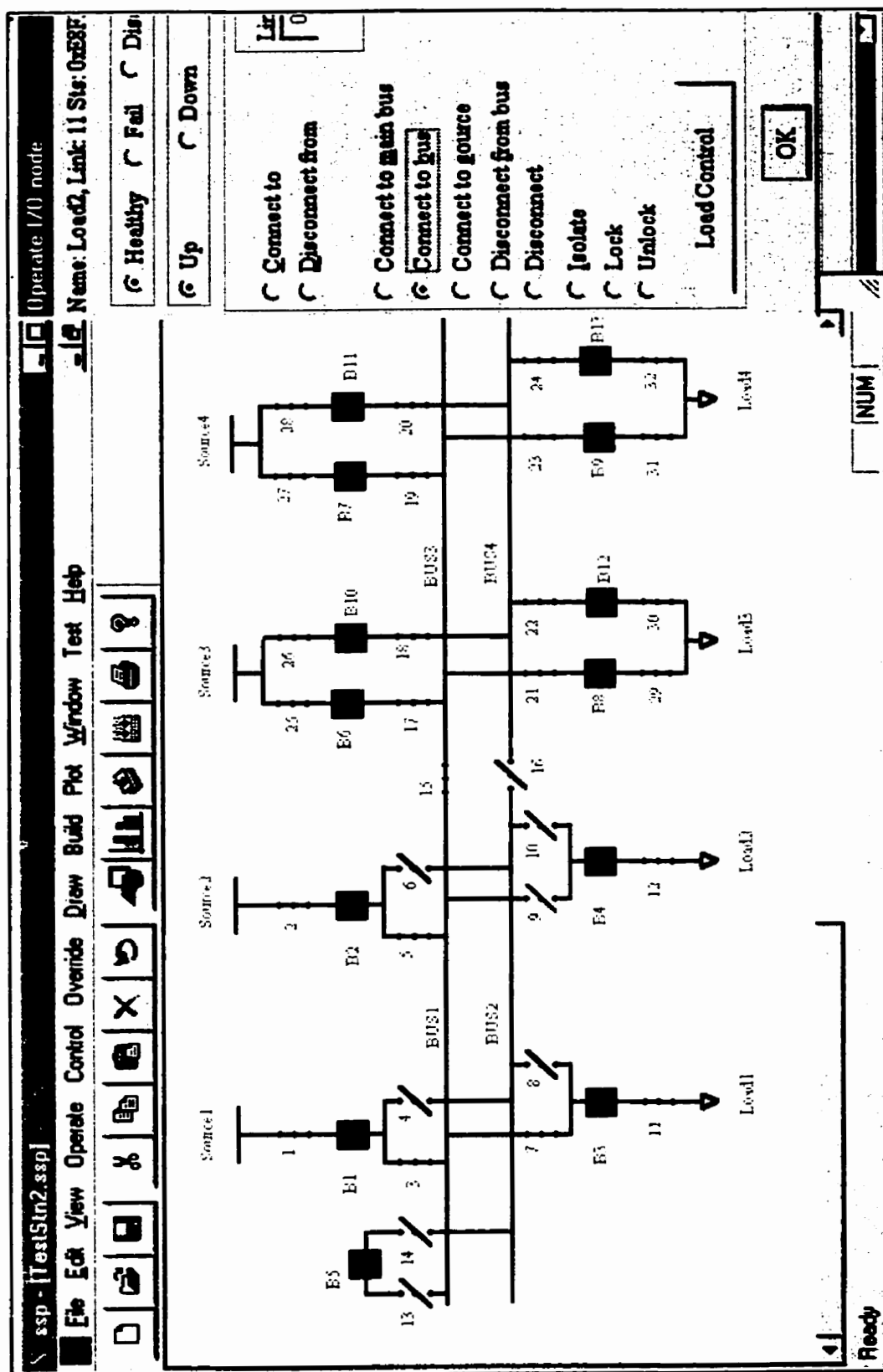


Figure H.49. Connecting Load2 to the busbar.

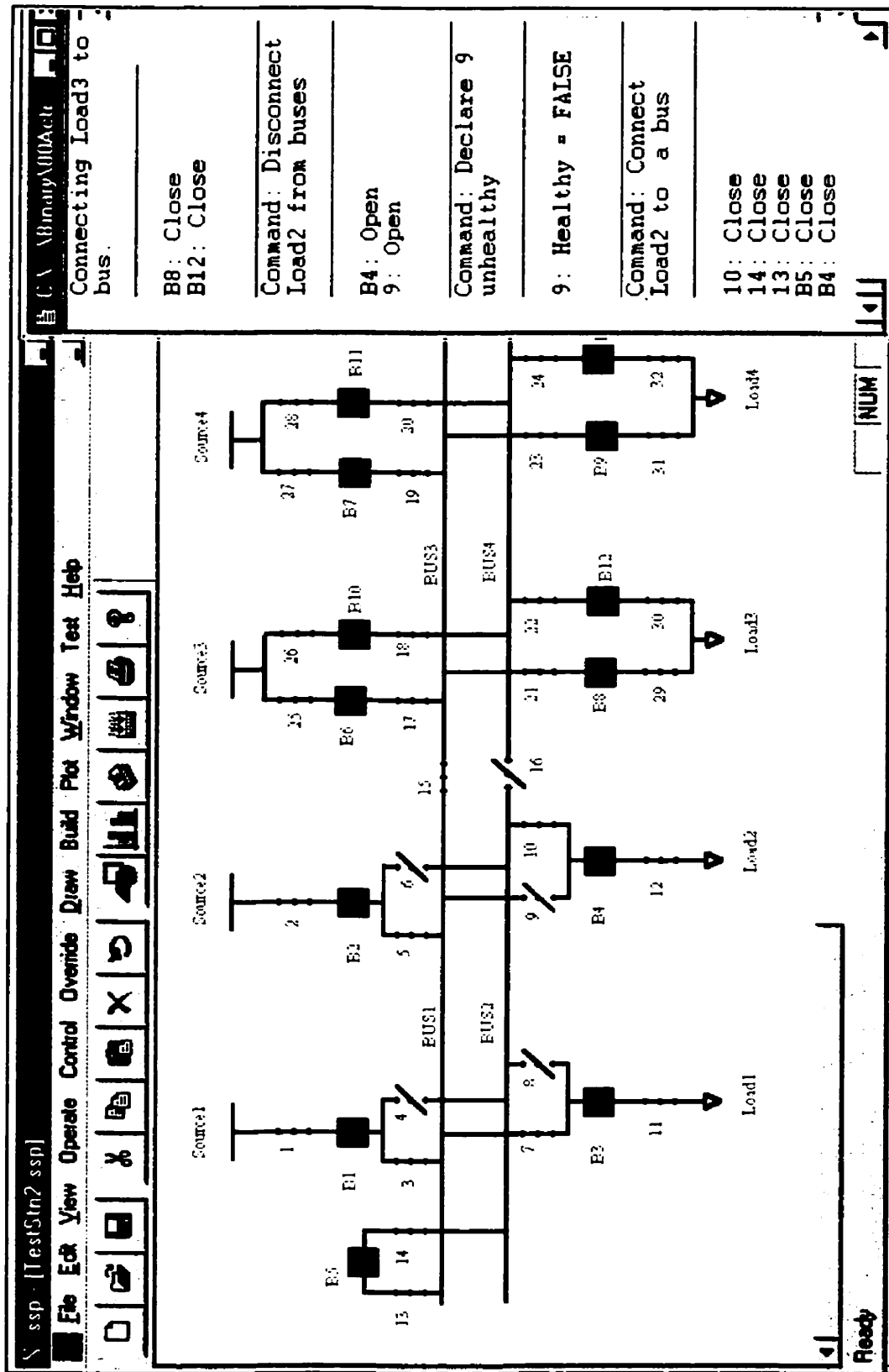


Figure H.50. Load2 is connected to Bus1 while avoiding switch 9.

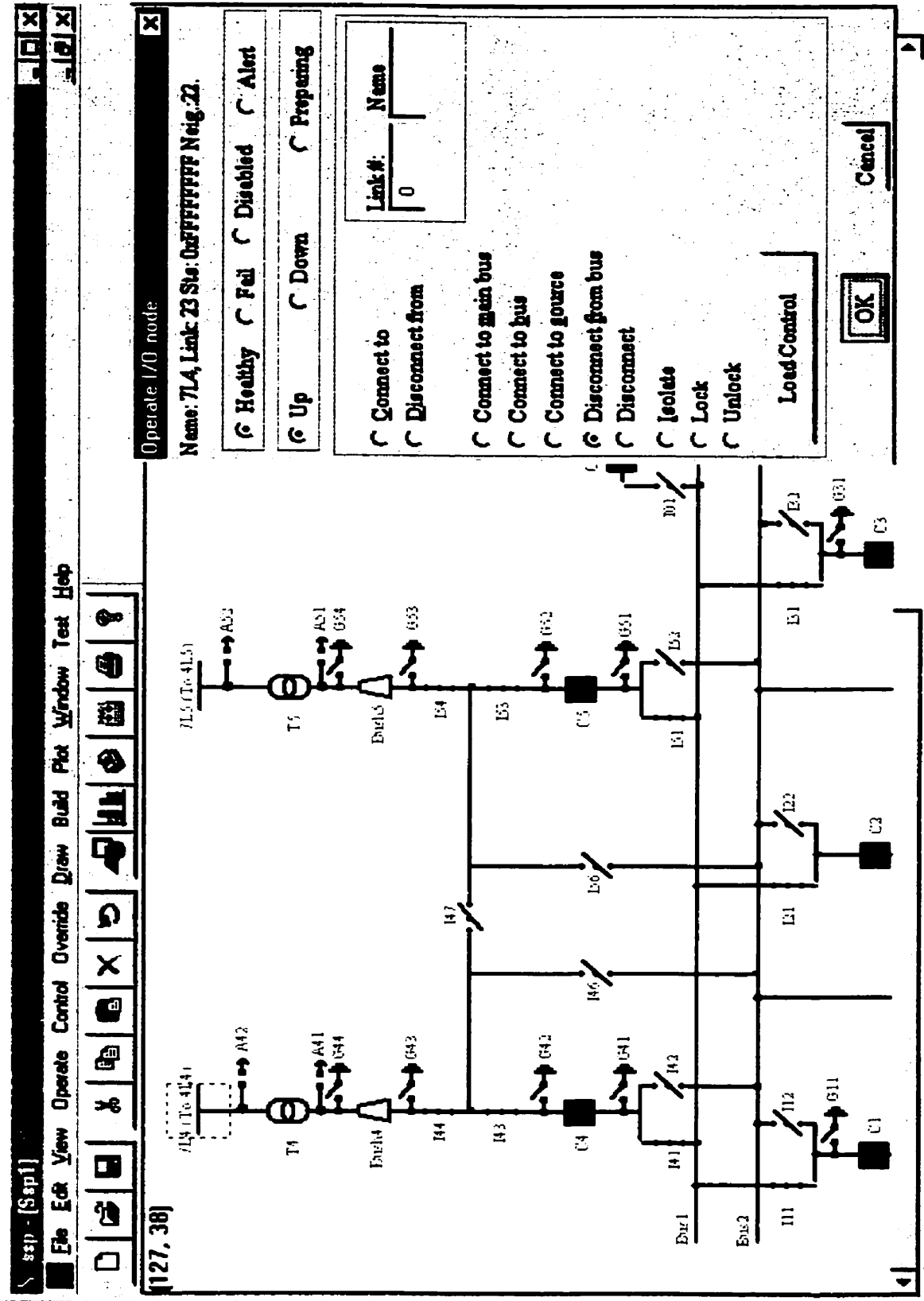
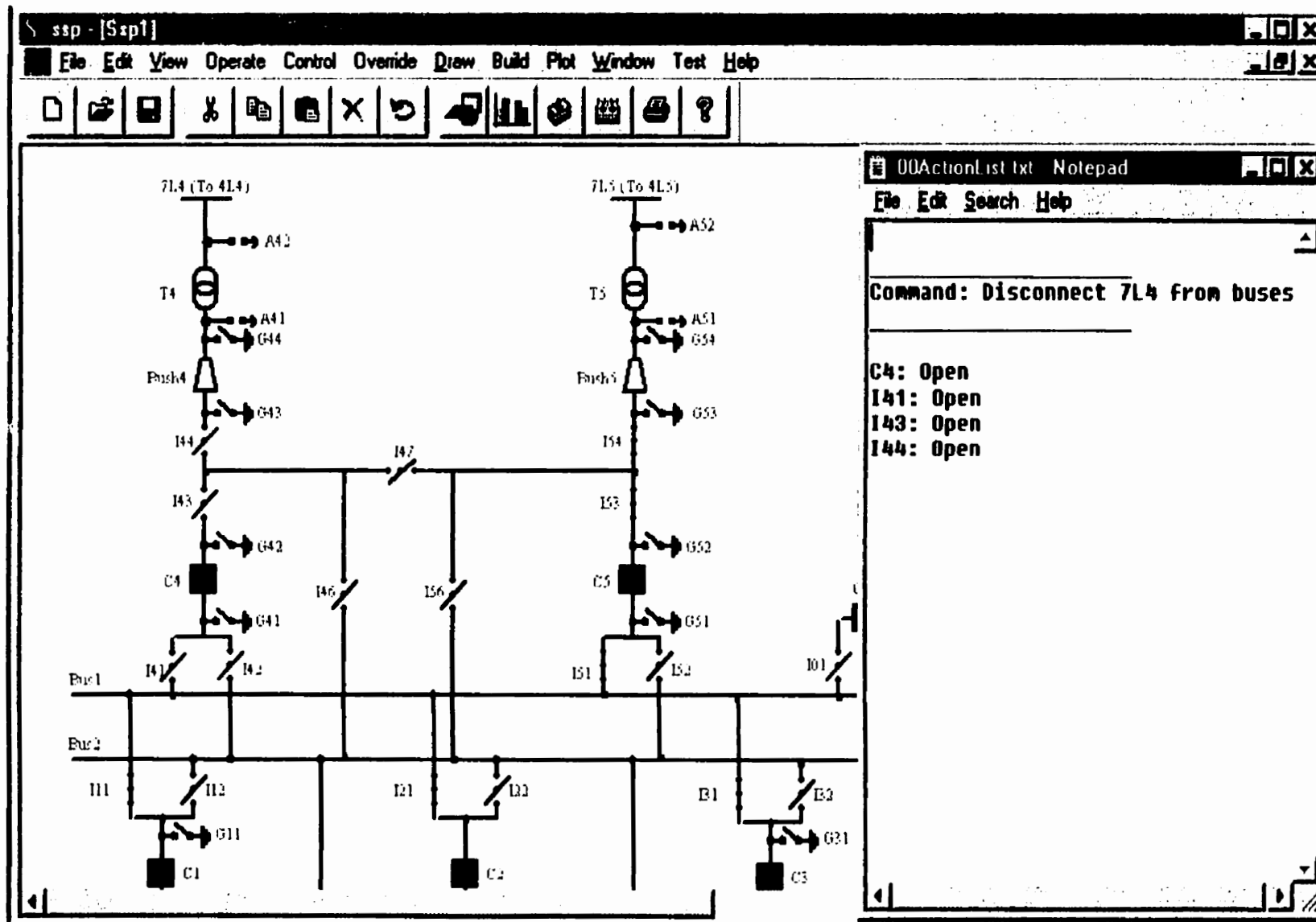


Figure H.51. Command being issued to disconnect line 7L4 from the busbars.





**Figure H.52.** The status of switches after the SSP disconnects line 7L4 from the busbars.

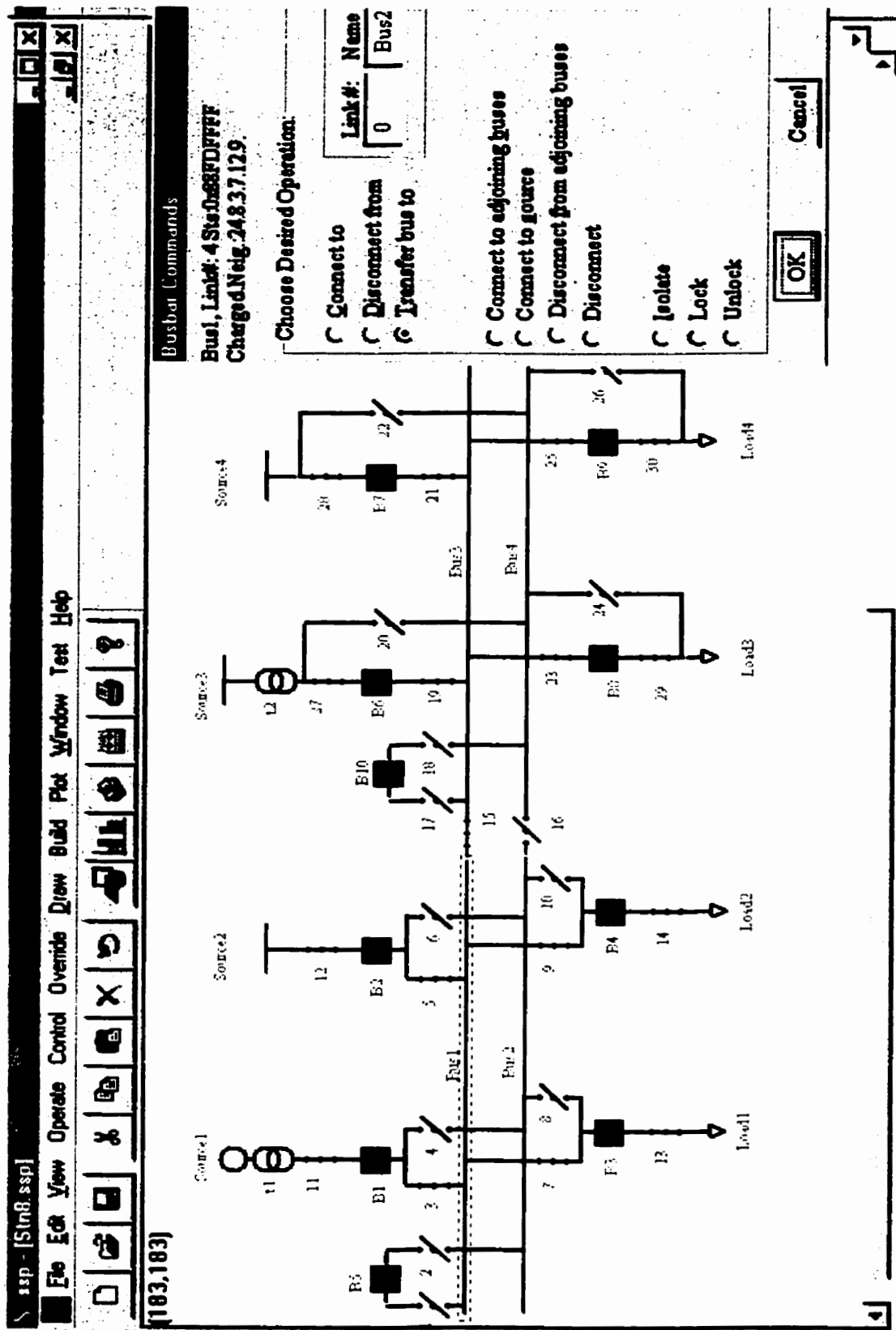


Figure H.53. Command being issued to transfer lines from Bus1 to Bus2.

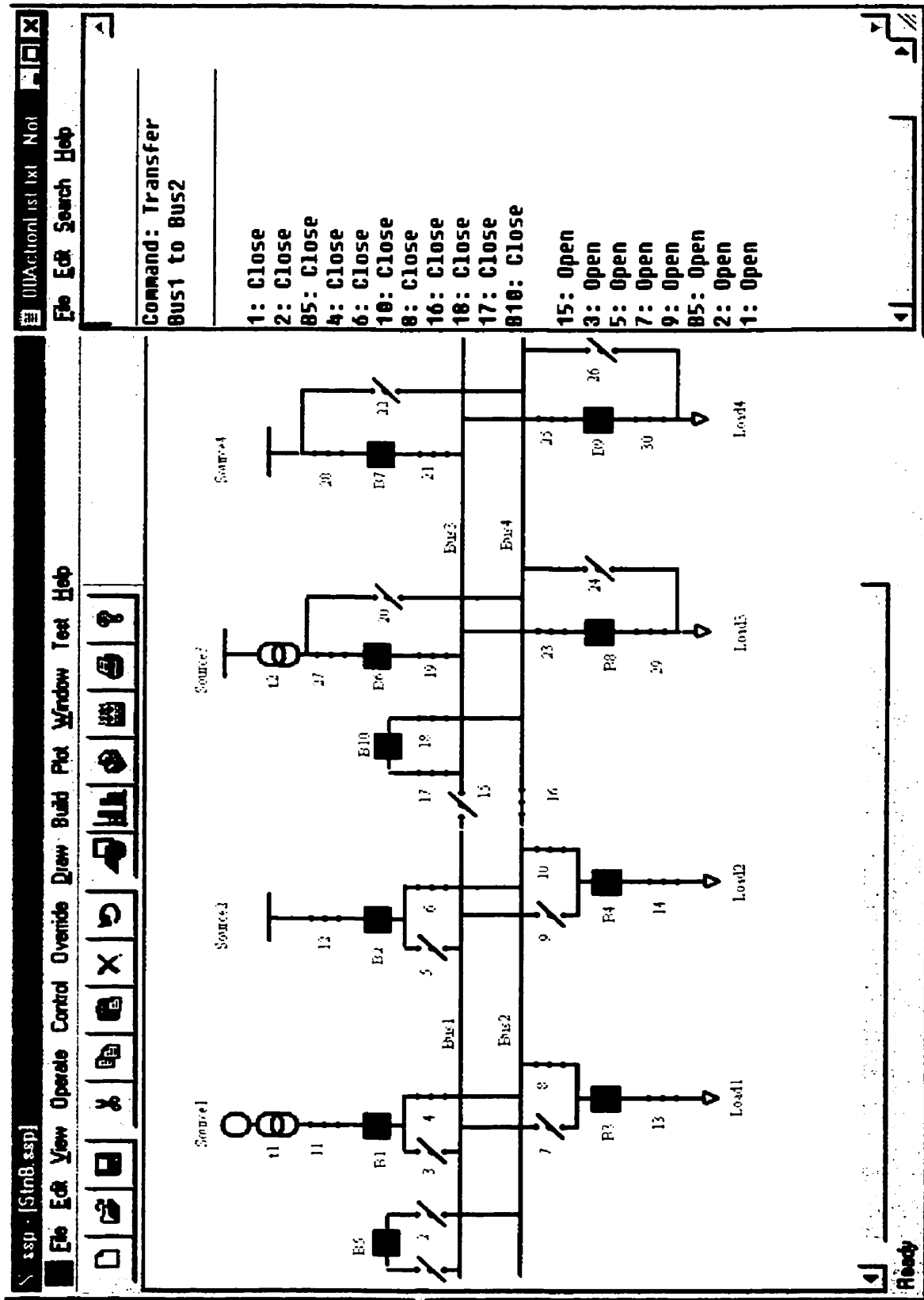


Figure H.54. The substation after the SSP transferred lines from Bus1 to Bus2.

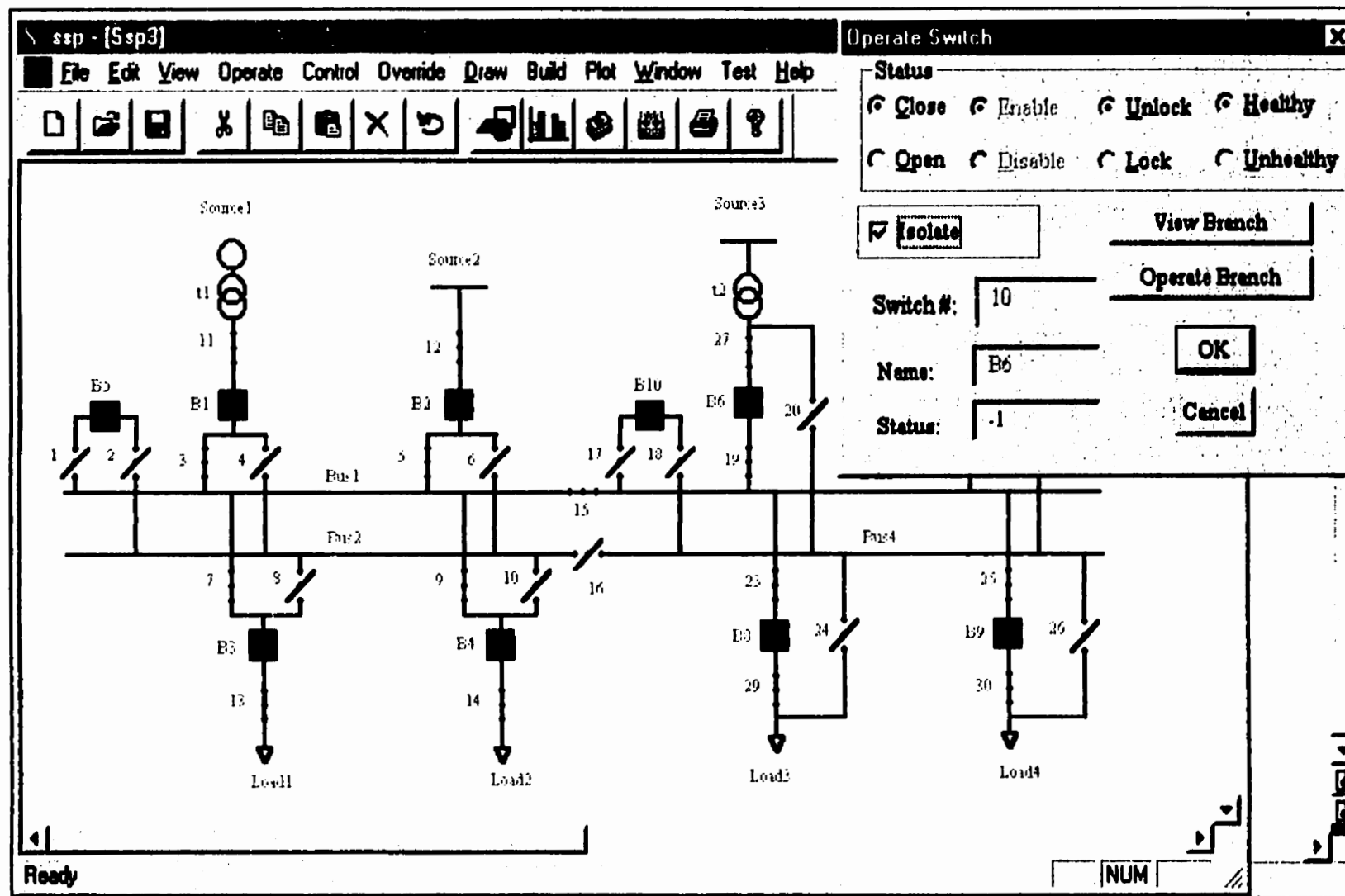
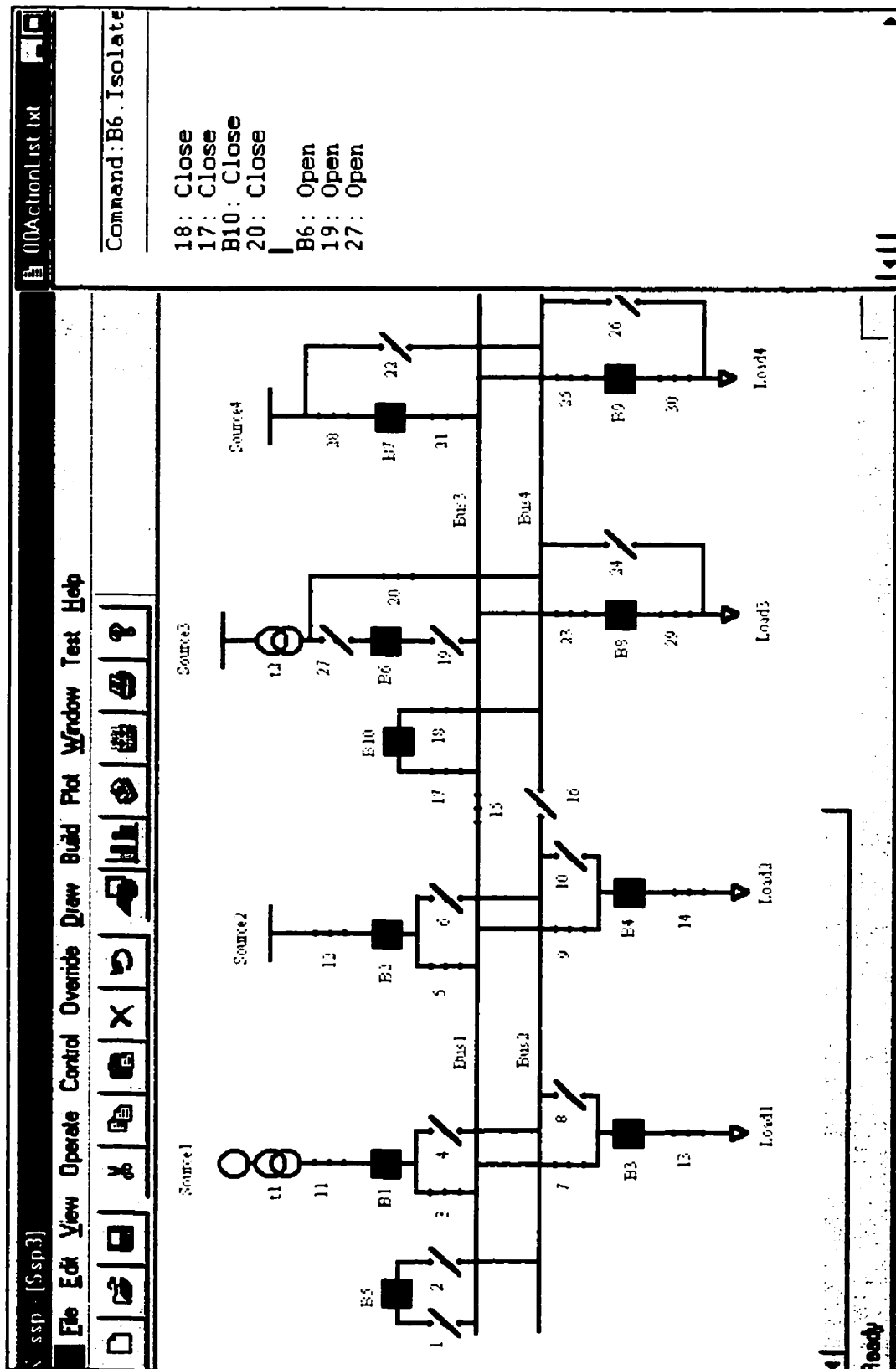


Figure H.55. Command being issued to isolate circuit breaker B6.



**Figure H.56.** The substation after the isolation of B6 and the switching sequence required to isolate it.

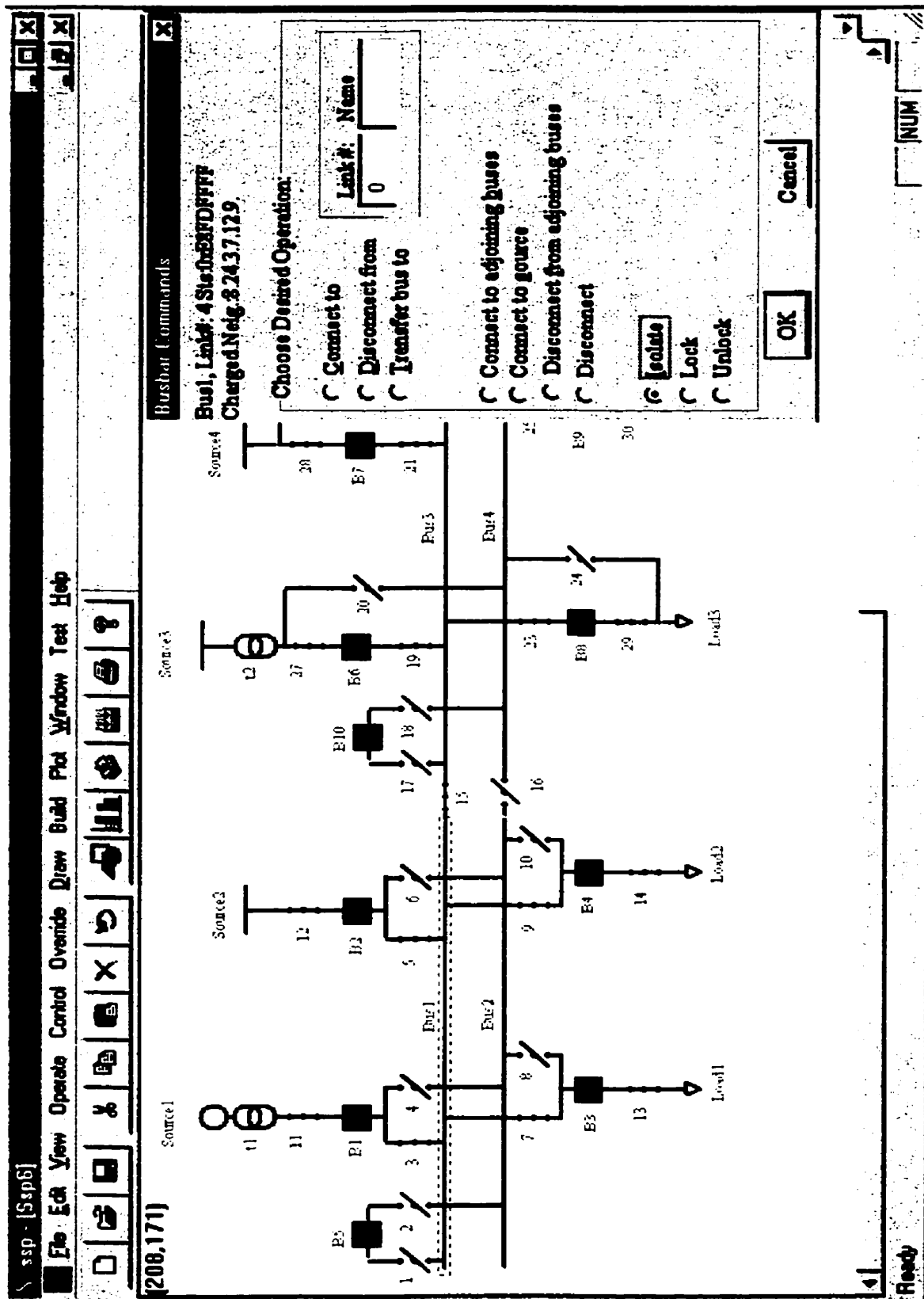


Figure H.57. Command being issued to isolate Bus1 on Test Substation 5.

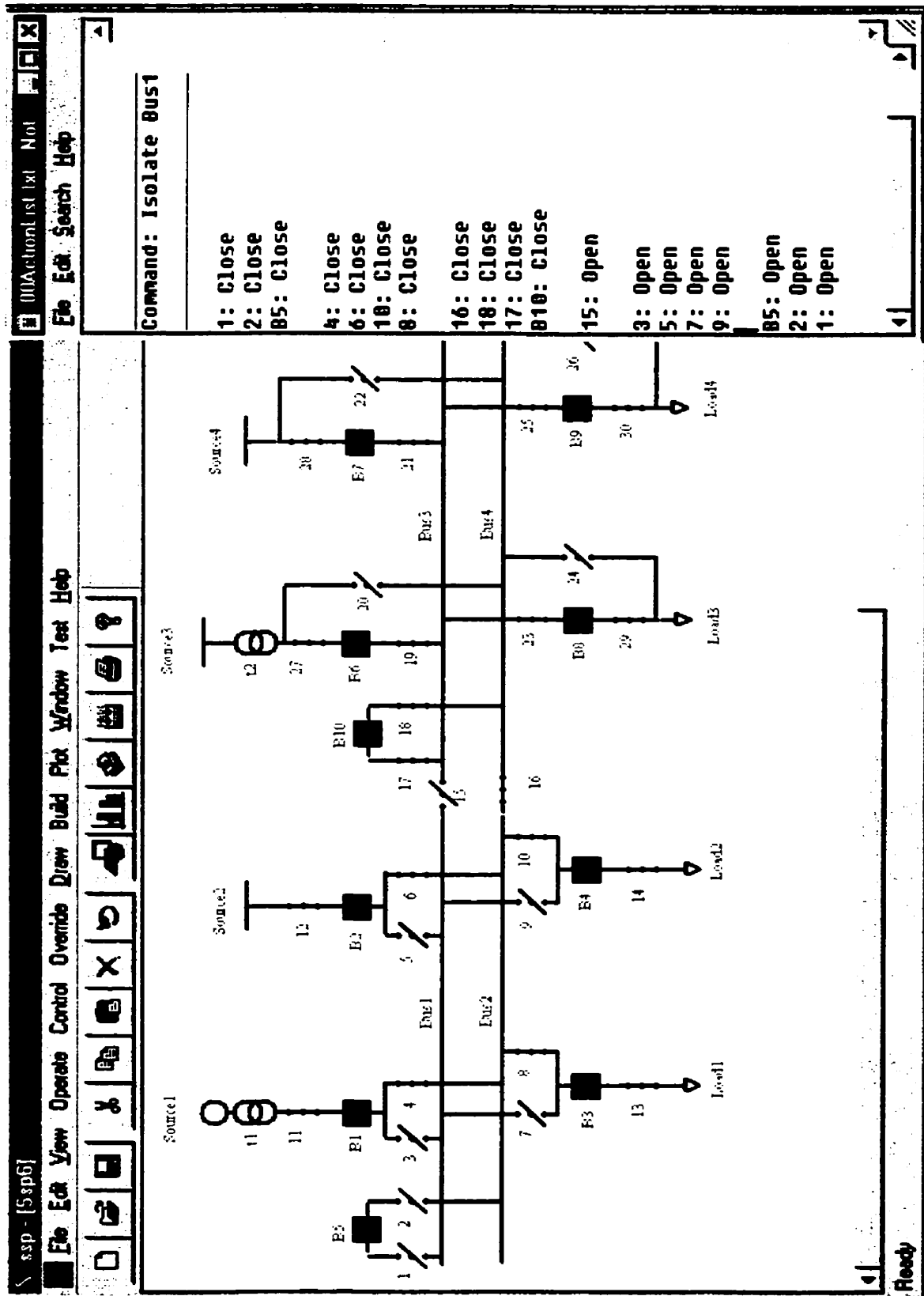


Figure H.58. The state of the substation after isolating Bus1.

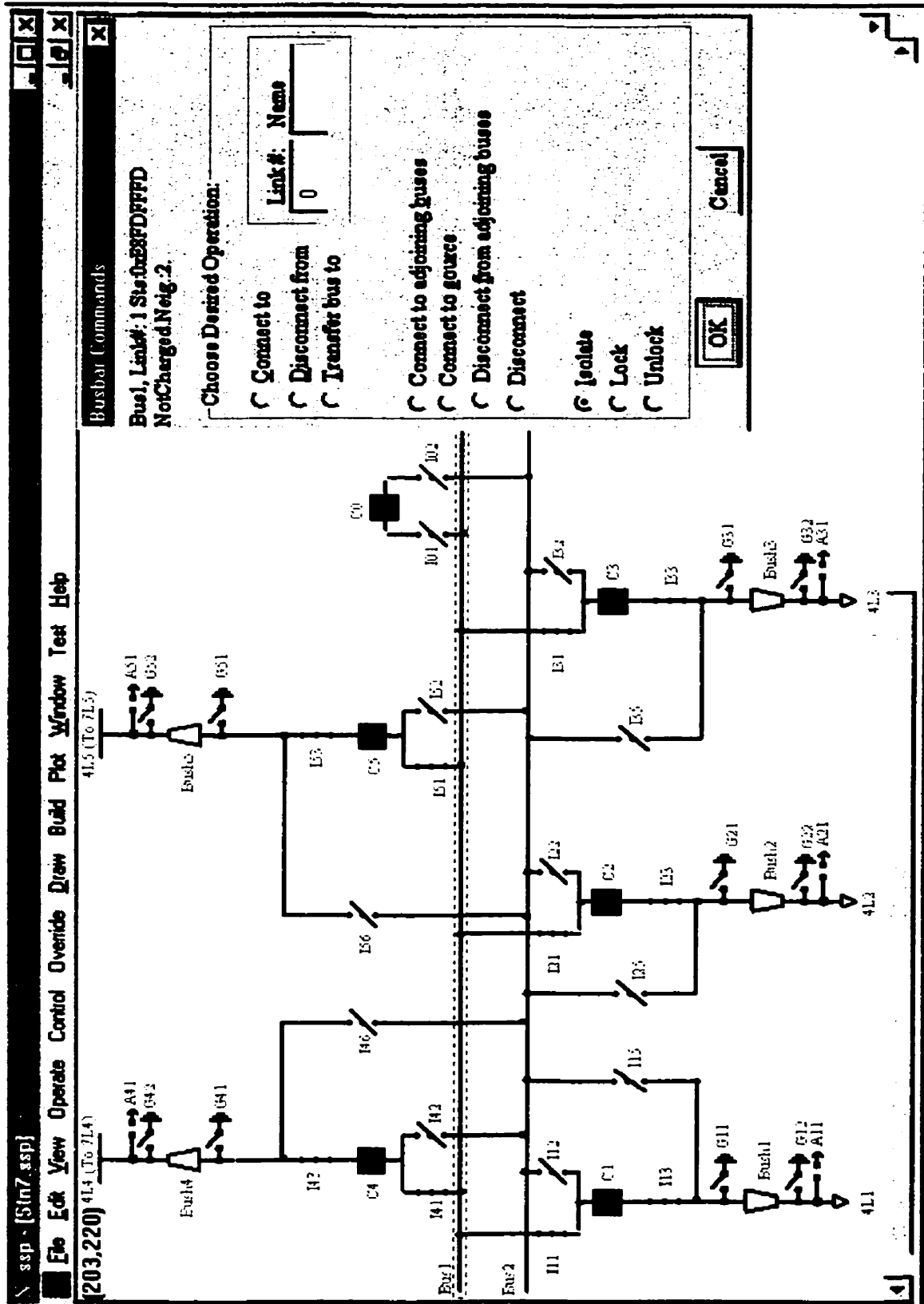
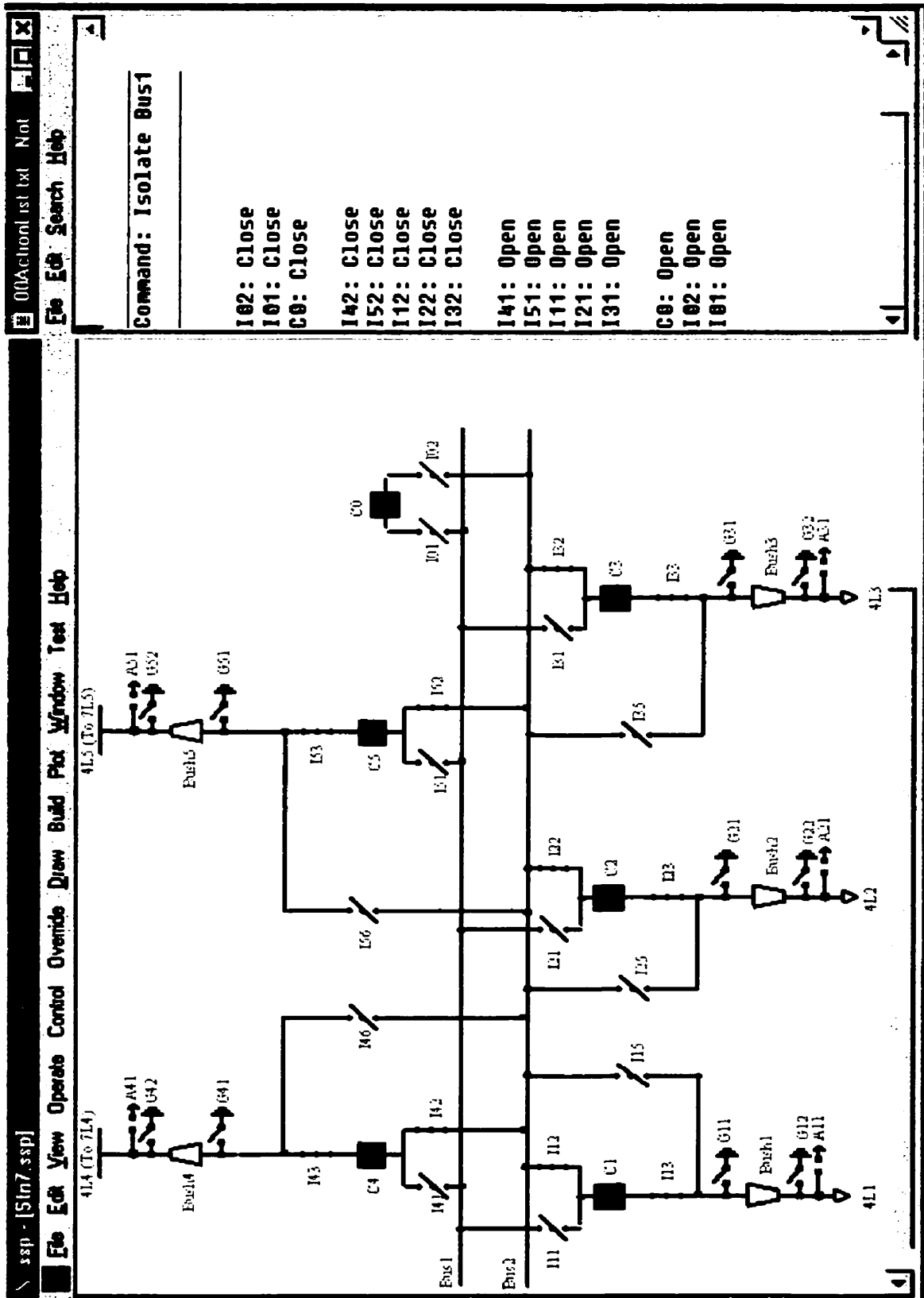


Figure H.59. Command being issued to isolate Bus 1.





**Figure H.60.** The status of the substation after isolating Bus1.

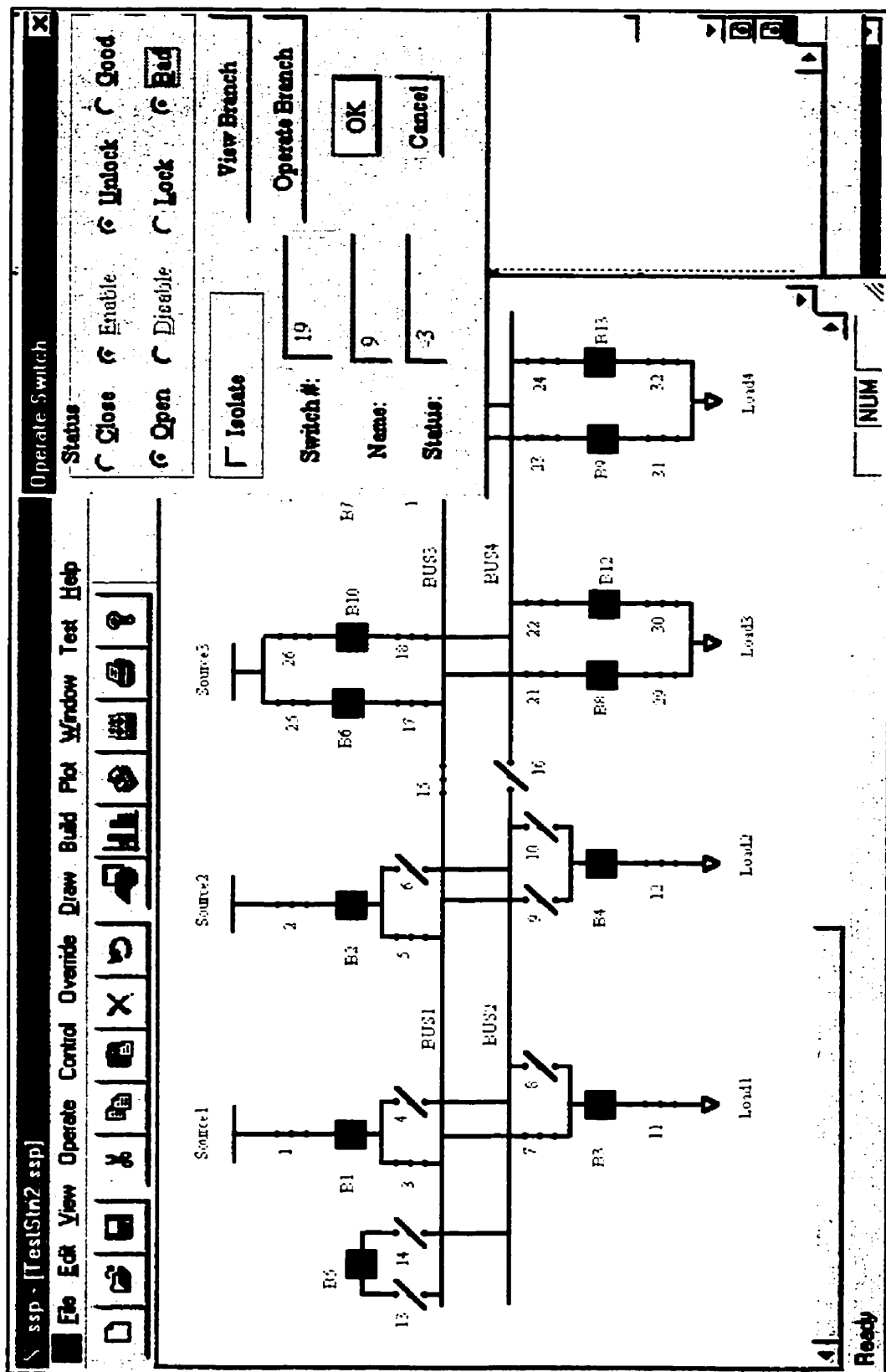


Figure H.61. Command being issued to declare isolator 9 as not healthy.

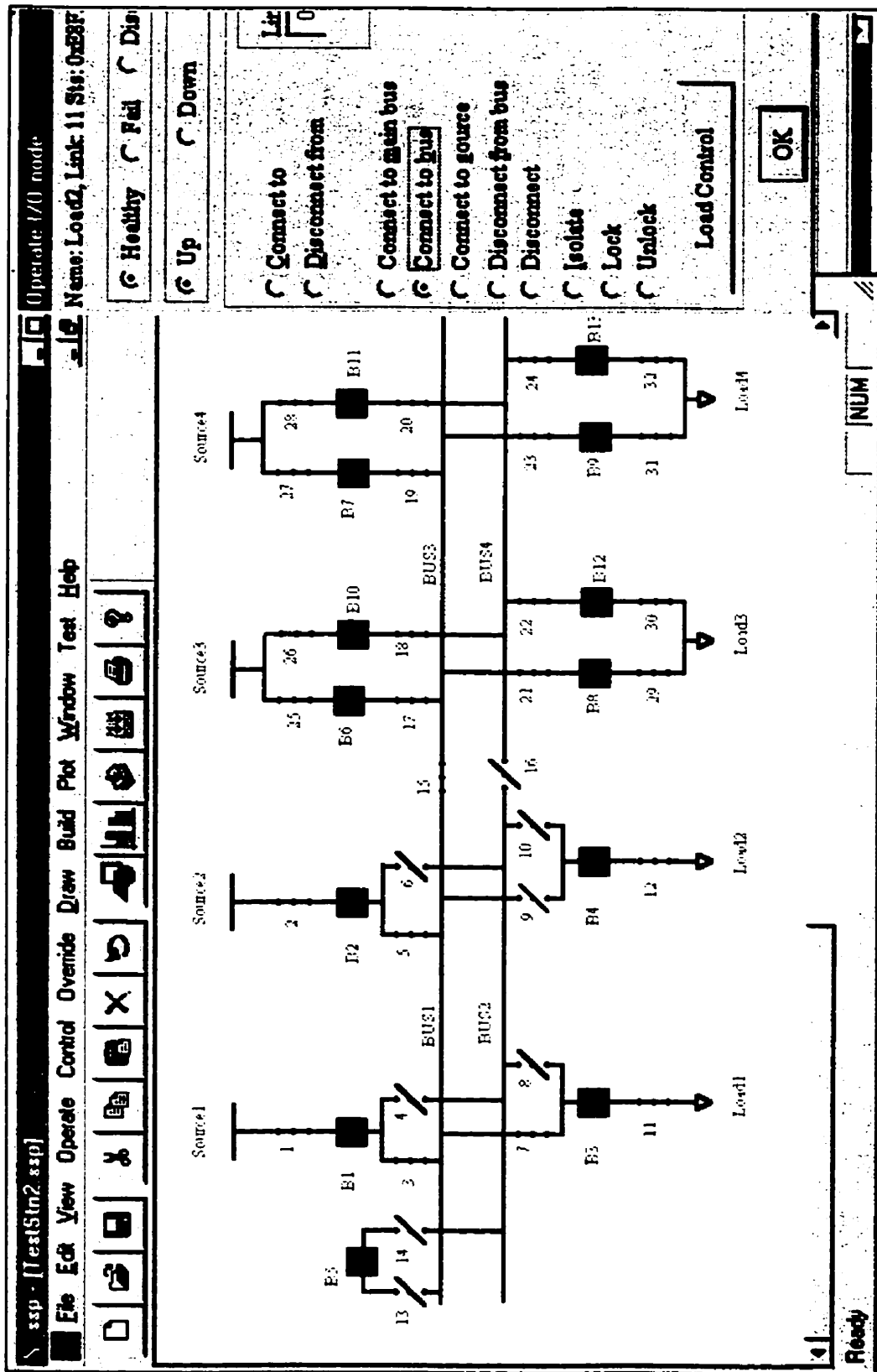


Figure H.62. The SSP has declared isolator 9 as not healthy and a command is being issued to connect Load2 to the busbar.

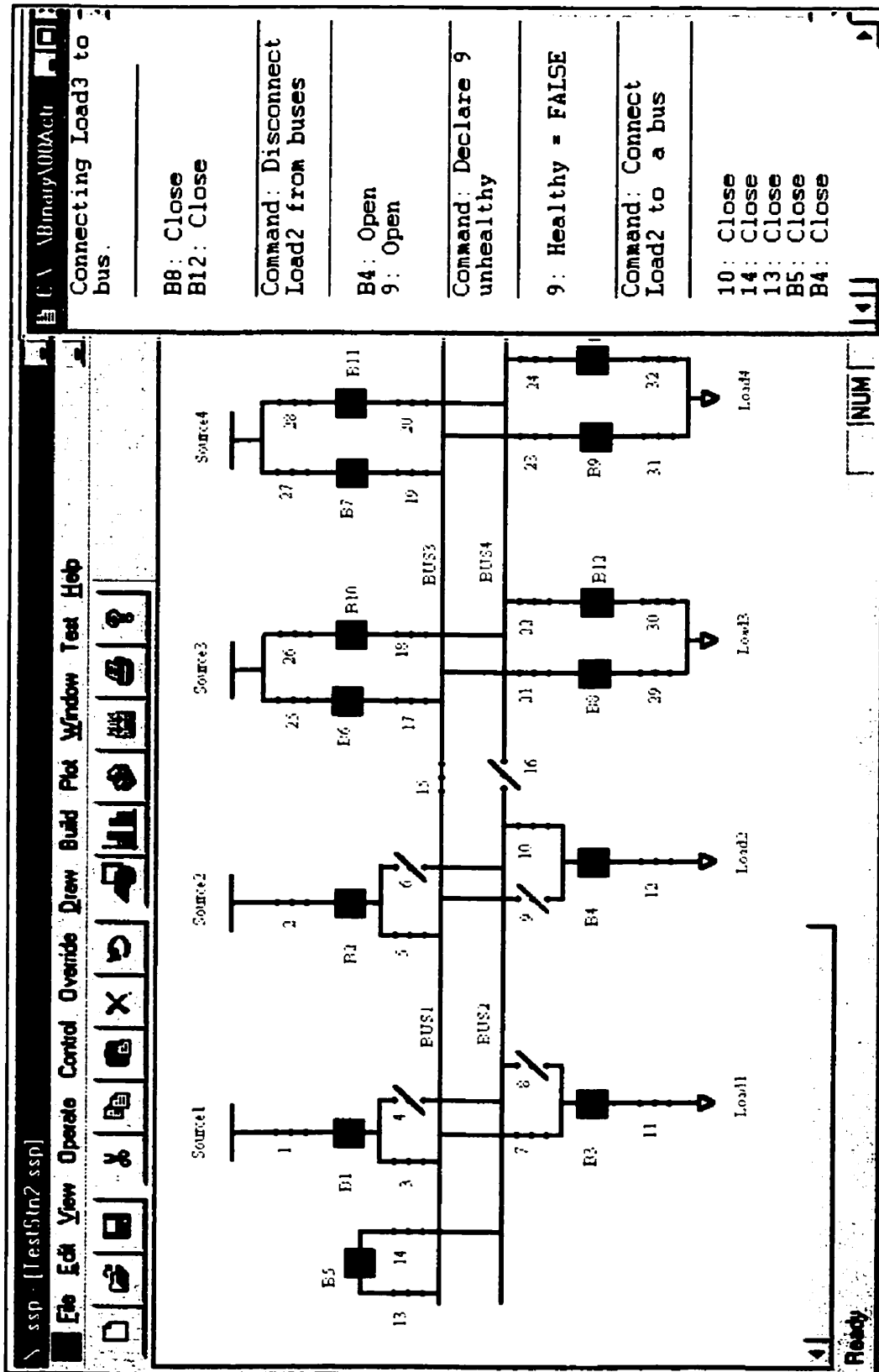


Figure H.63. The substation after the SSP connects Load2 to Bus1 while avoiding isolator 9.

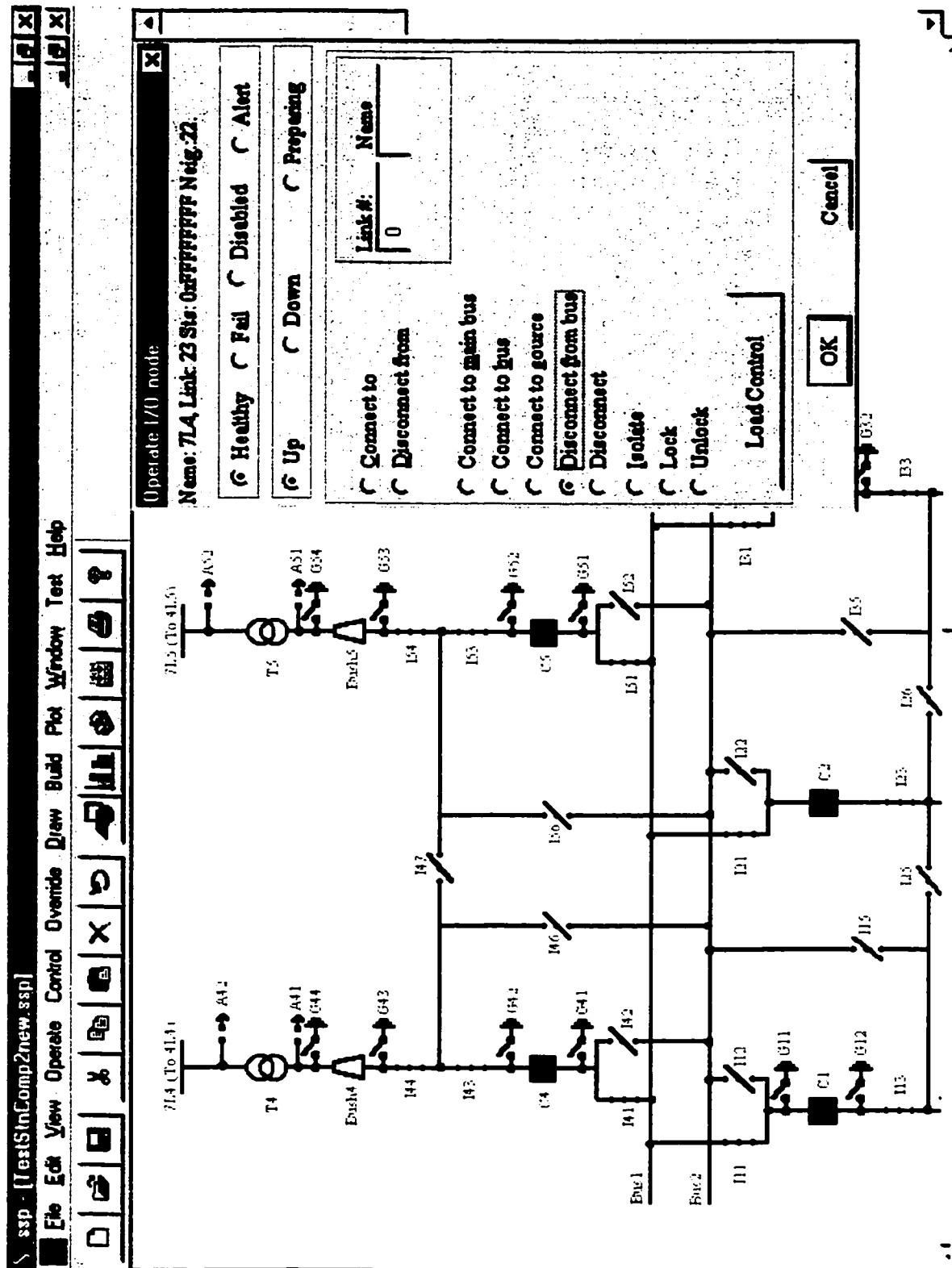
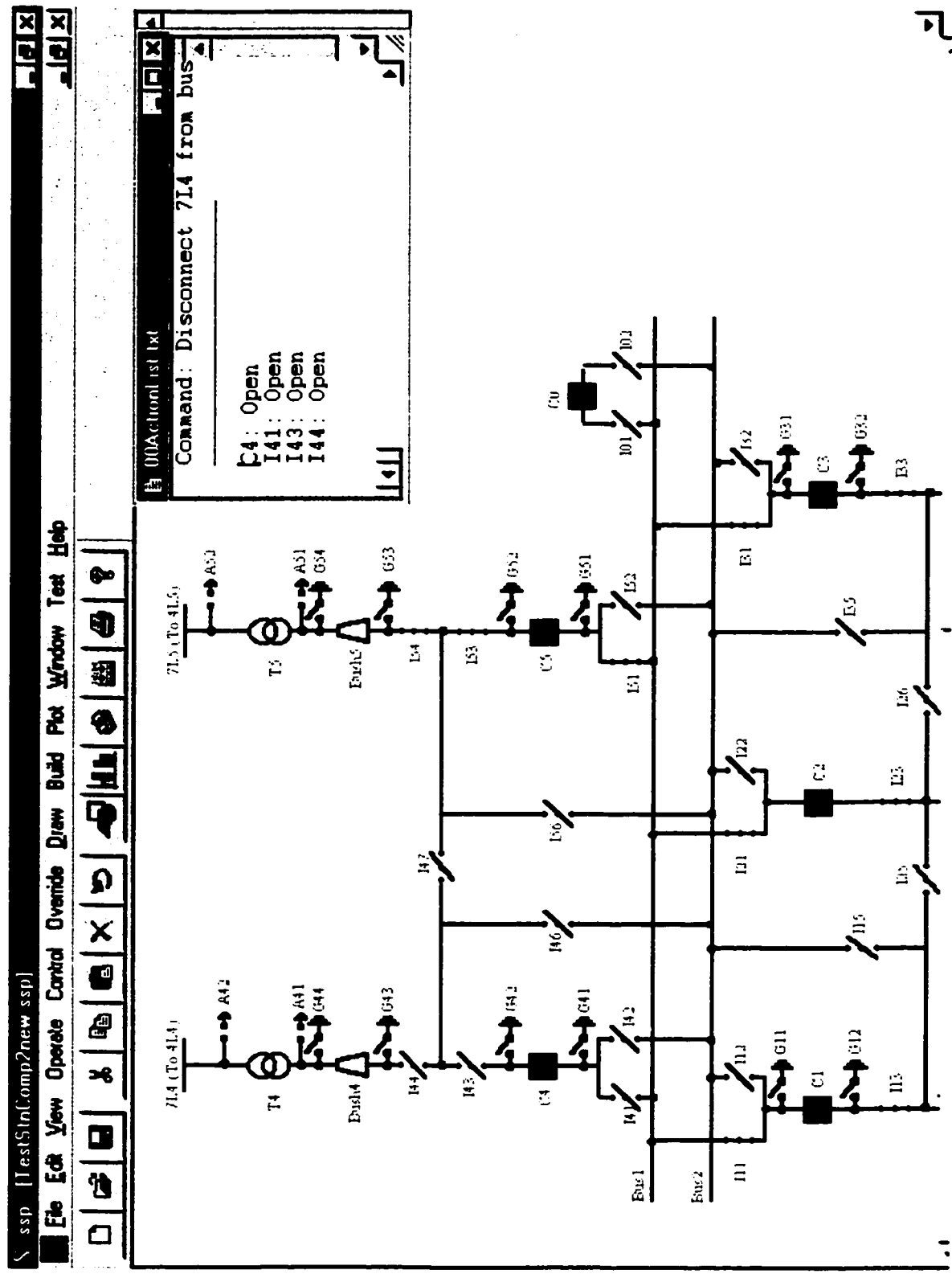
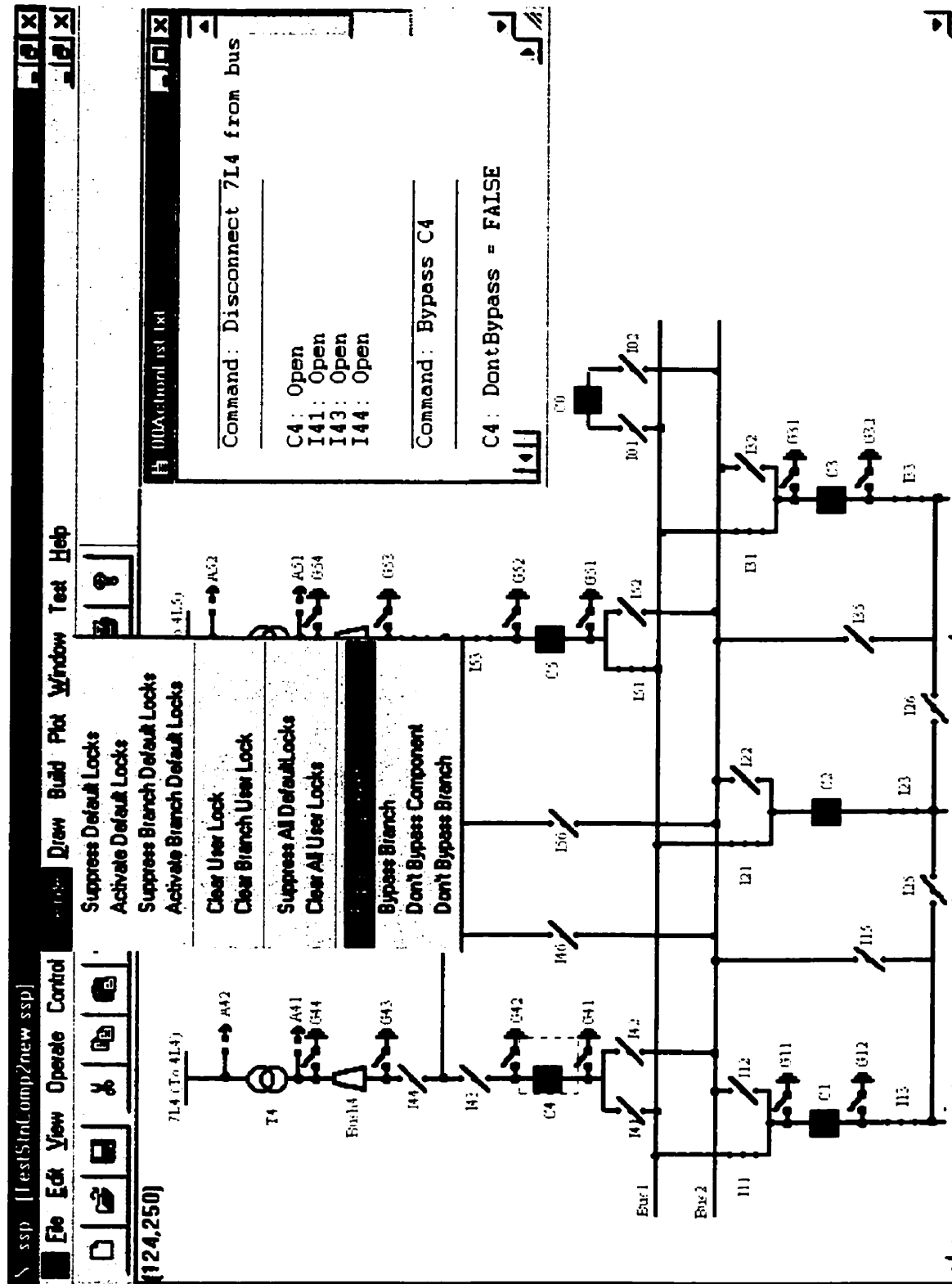


Figure H.64. Command being issued to disconnect source 7L4 from the busbar.



**Figure H.65.** Switching and final configuration of the substation after disconnecting line 7L4.



**Figure H.66.** Command being issued to bypass circuit breaker C4 from power flow paths to be formed in future.

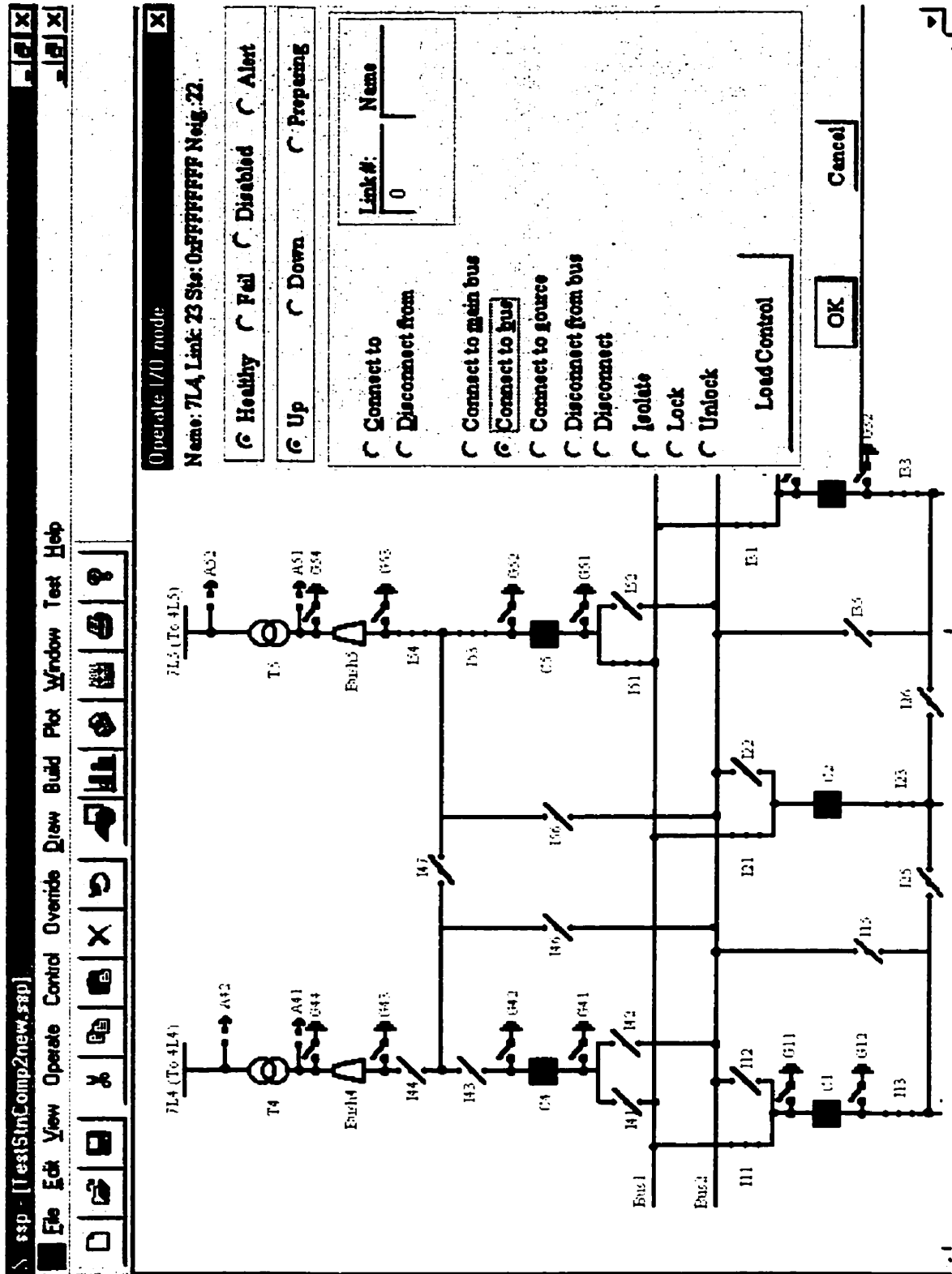


Figure H.67. Command being issued to connect line 7L4 to the busbar after bypass C4.



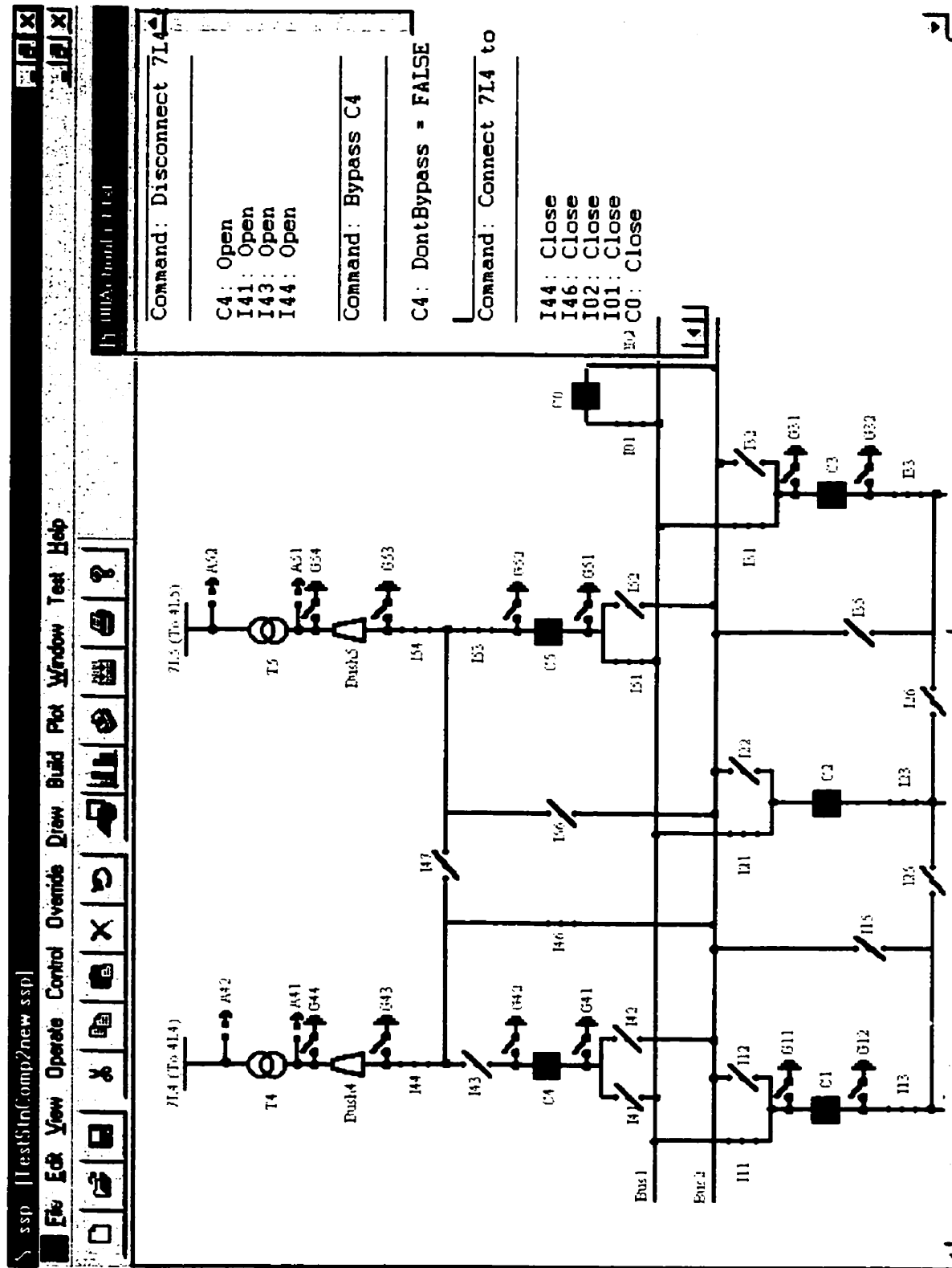
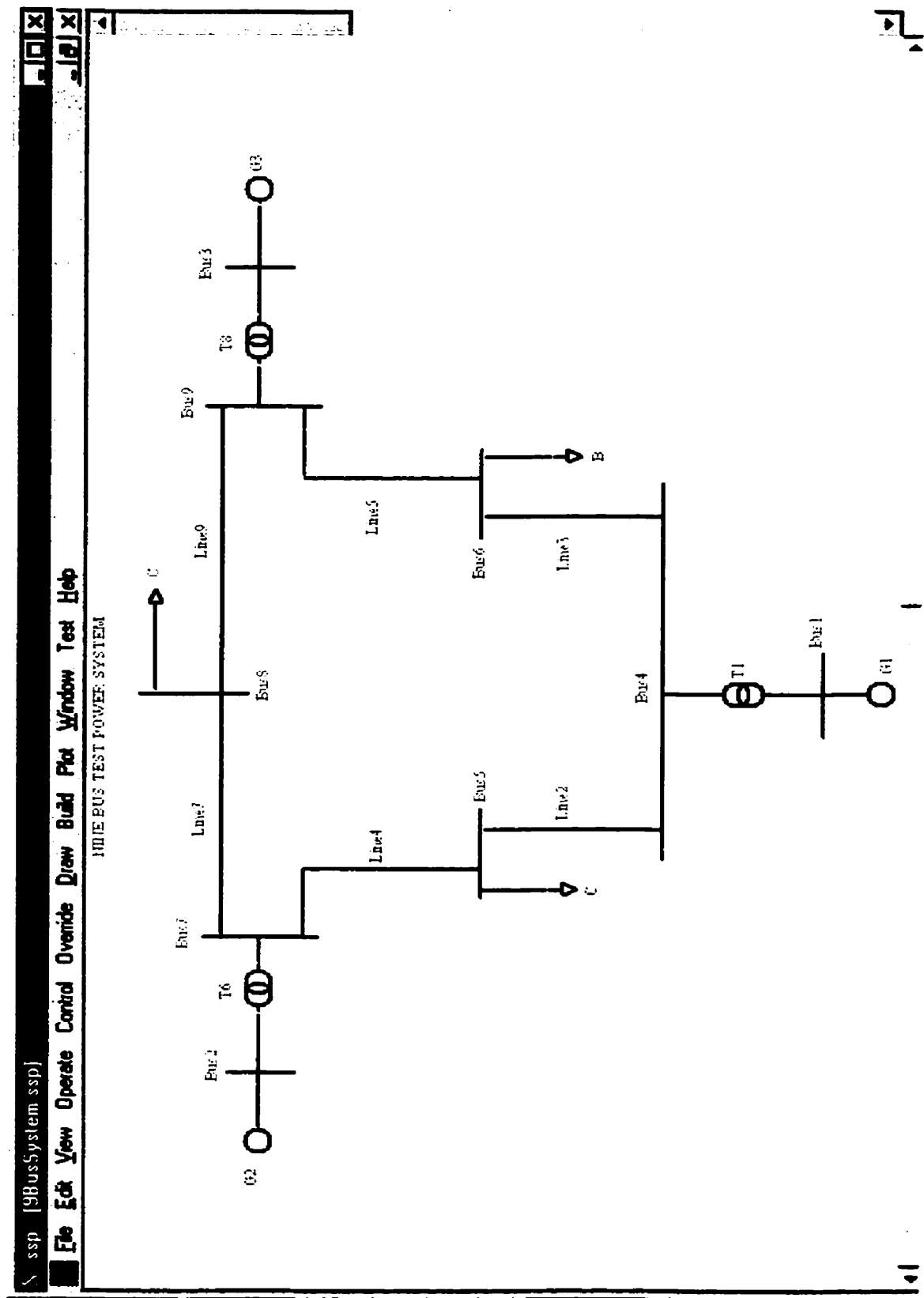


Figure H.68. Connection made between line 7L4 and Bus1 while bypassing C4.



**Figure H.69.** Nine bus power system whose Bus3 represents the substation under test.

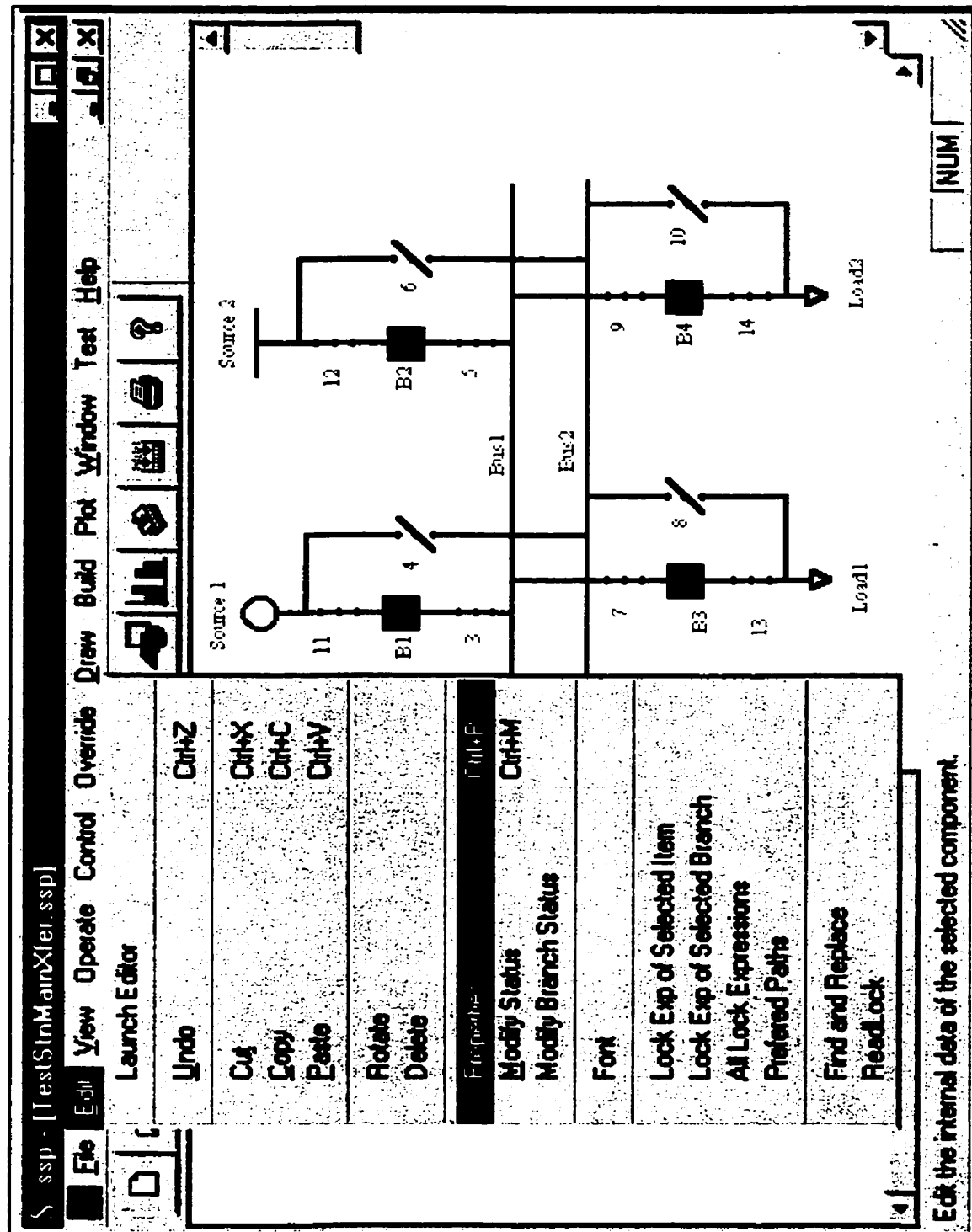


Figure H.70. Command being issued to edit data that relate the substation with the power system.

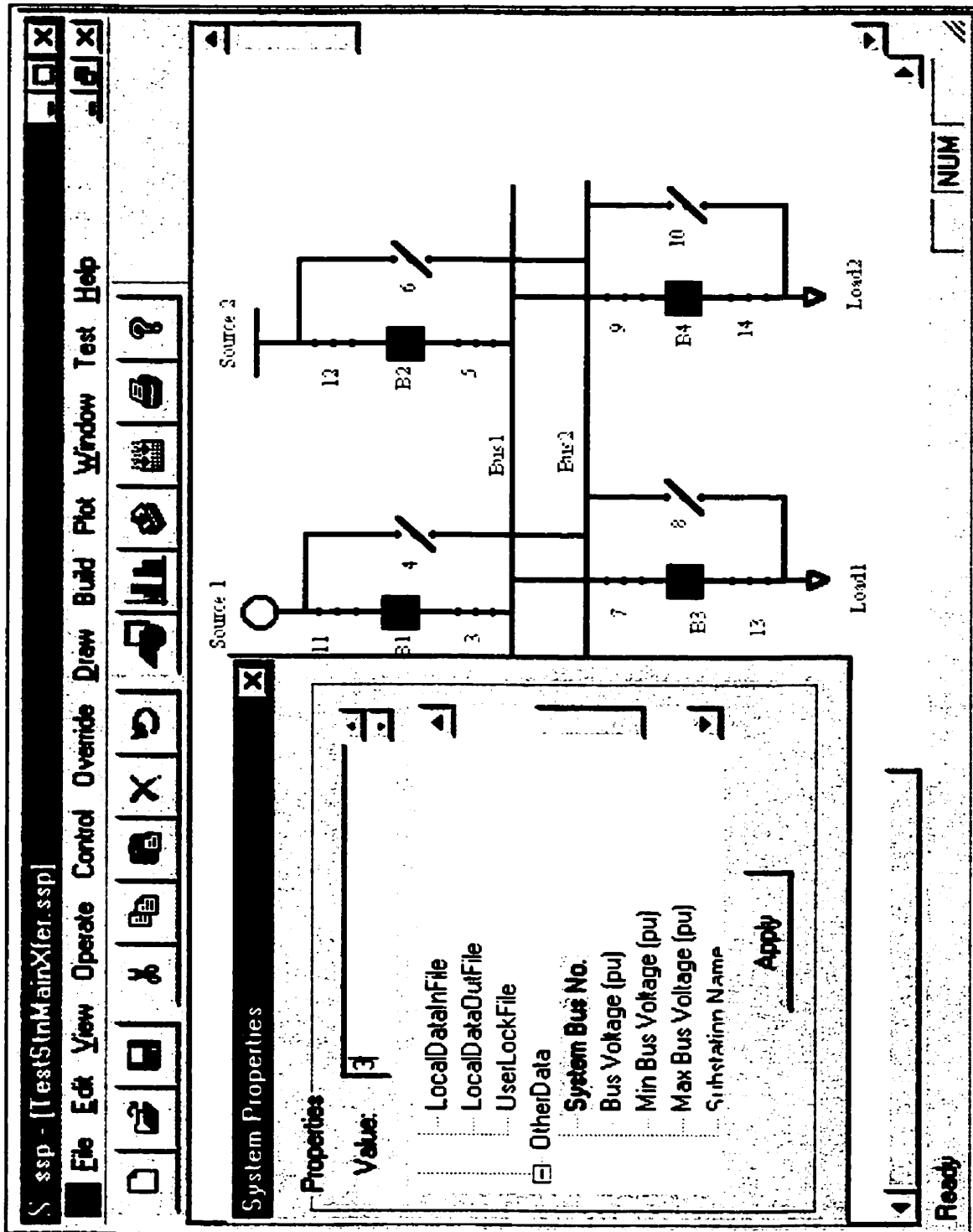


Figure H.71. The system bus number corresponding to the substation is being specified.

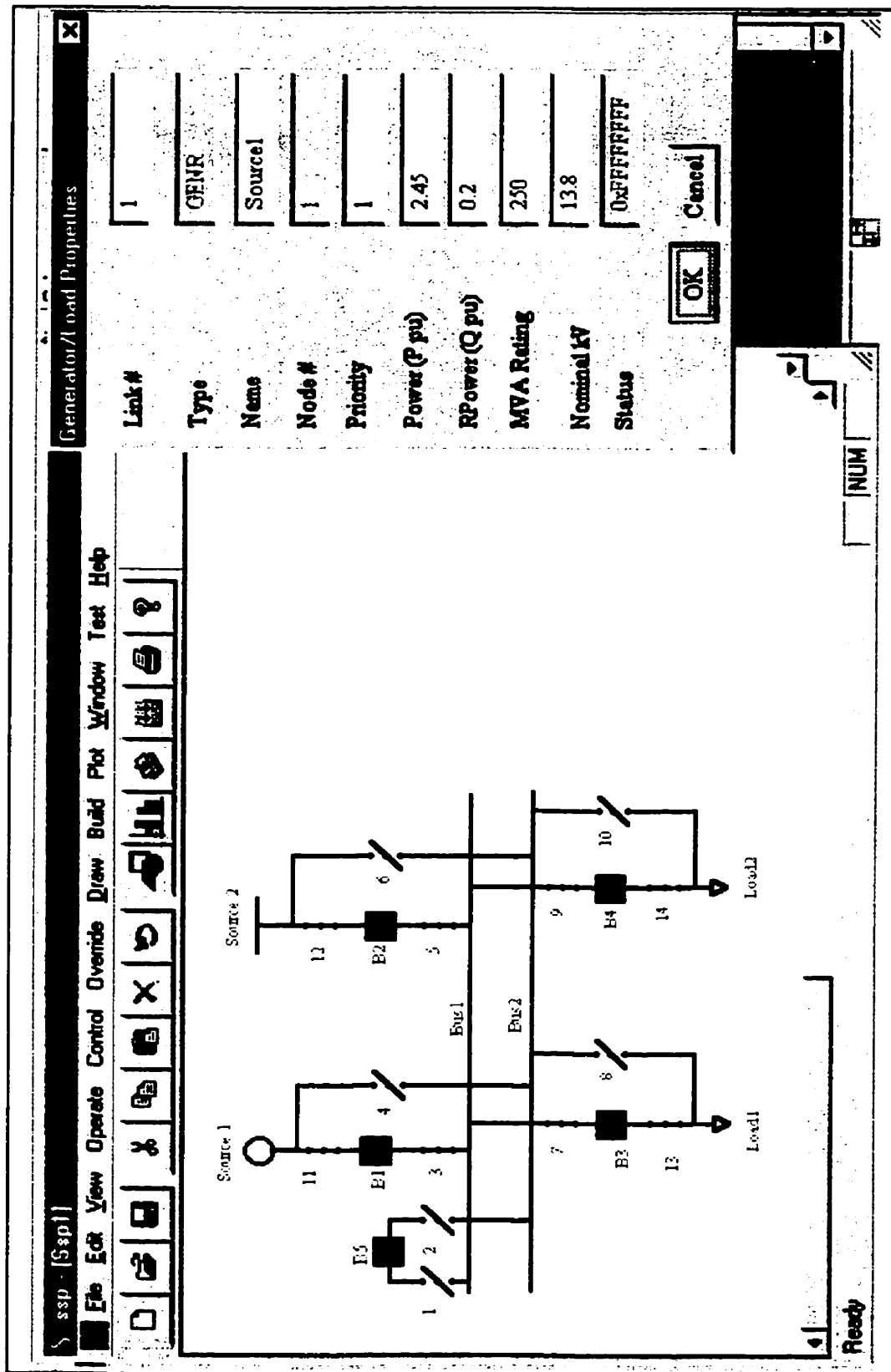


Figure H.72. Load and Generator data being specified.

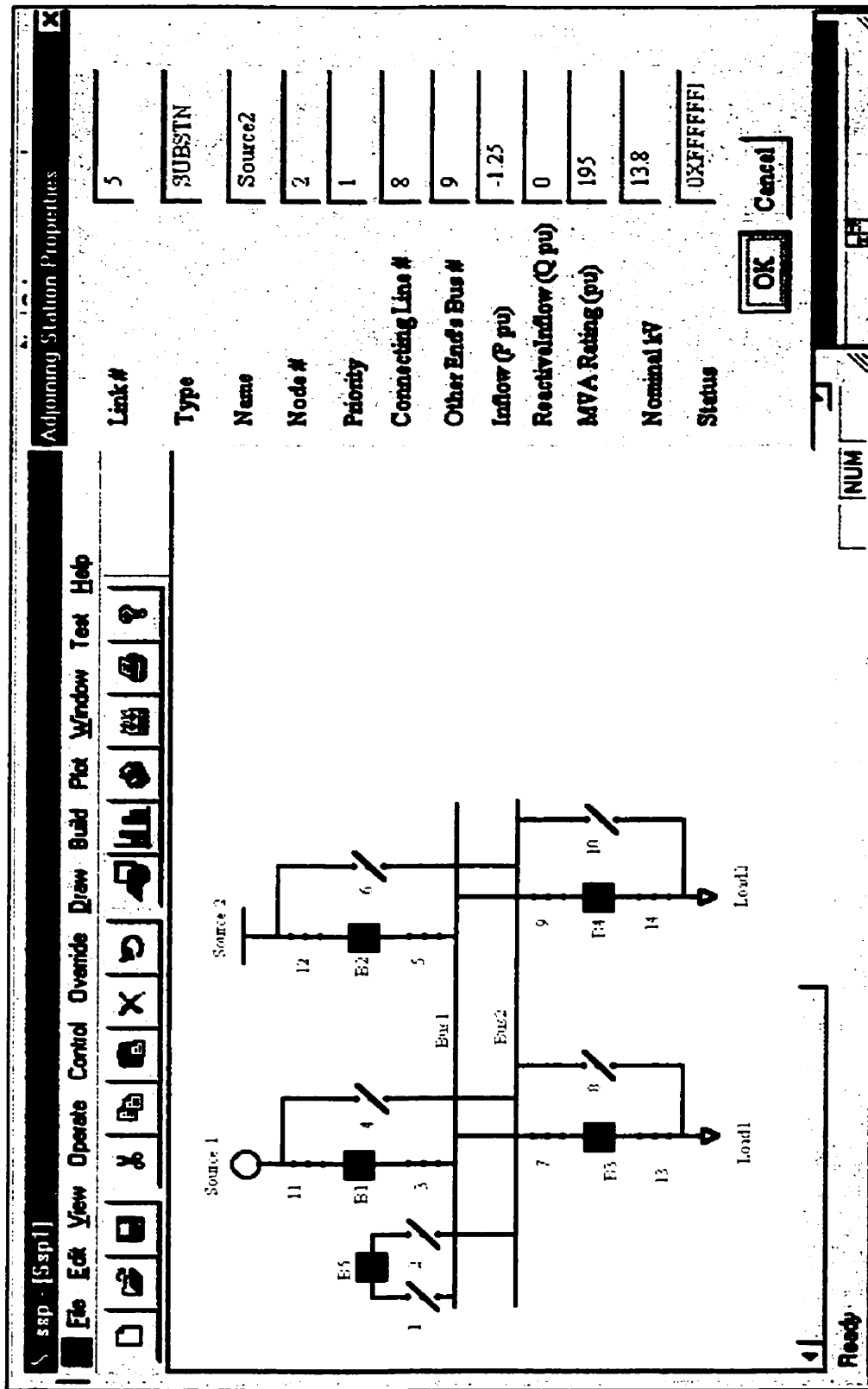


Figure H.73. Properties of node connecting to the rest of the power system being specified.

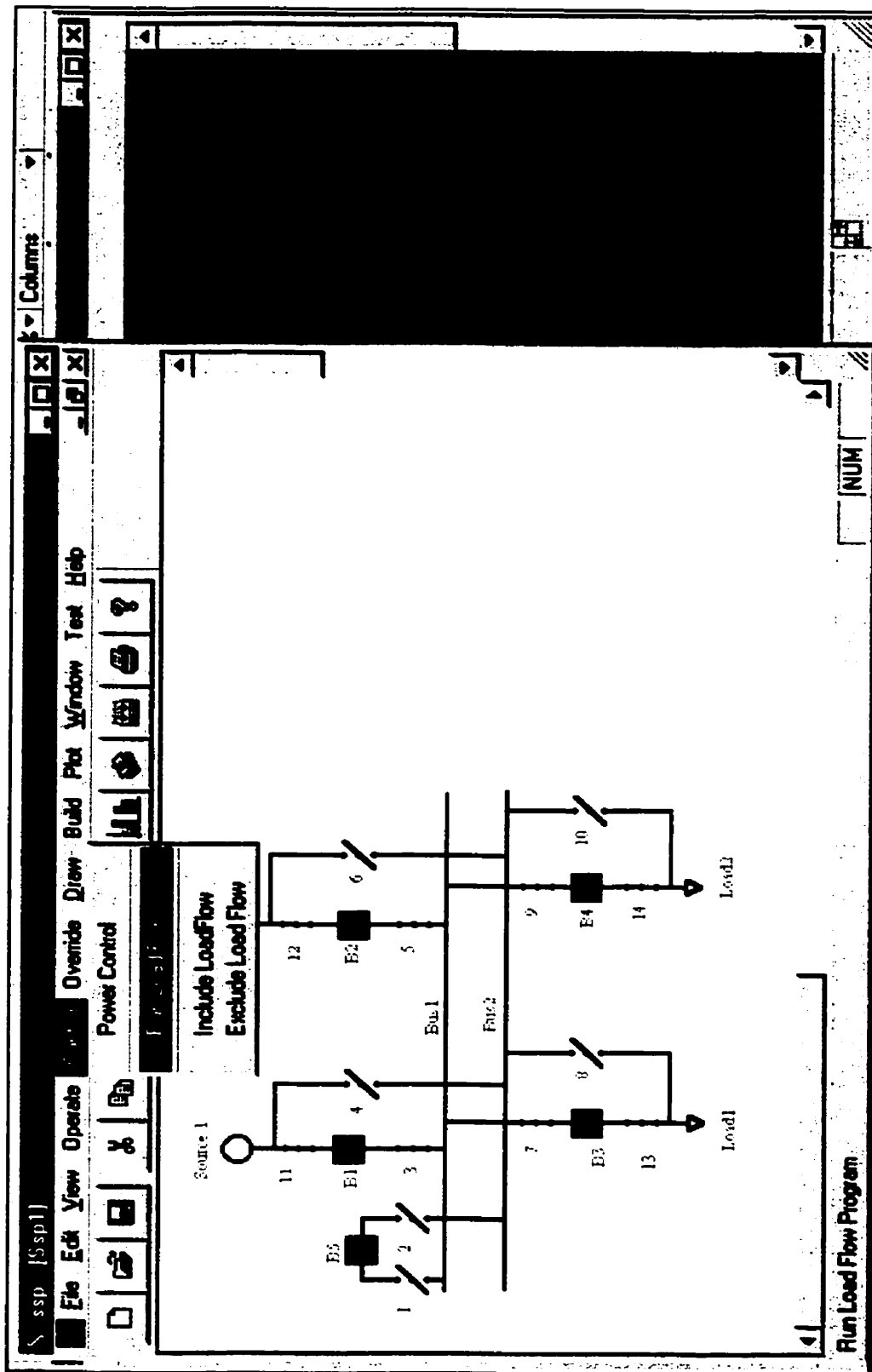


Figure H.74. Load-flow program being launched.

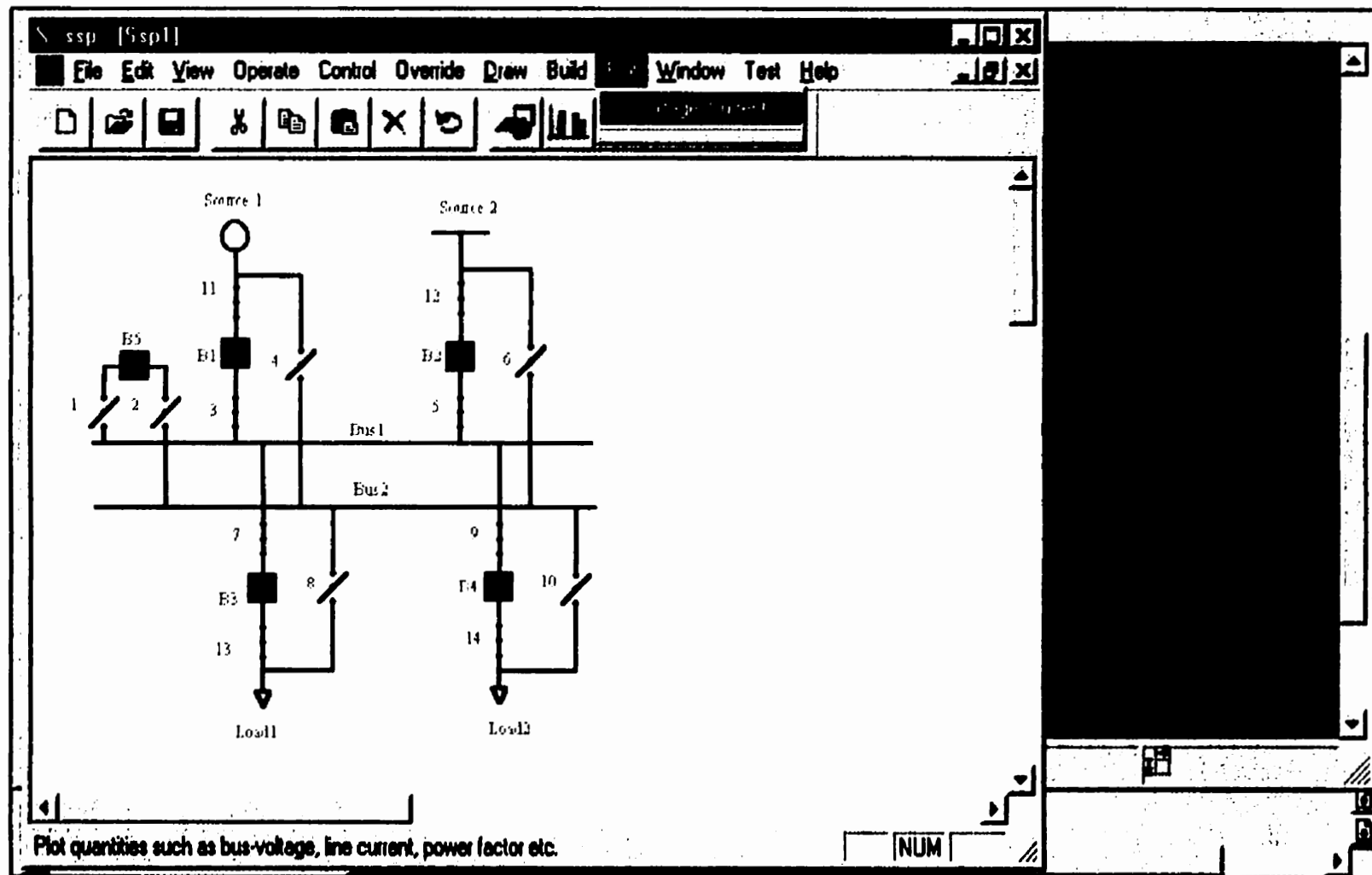


Figure H.75. Graph plotting program being launched.



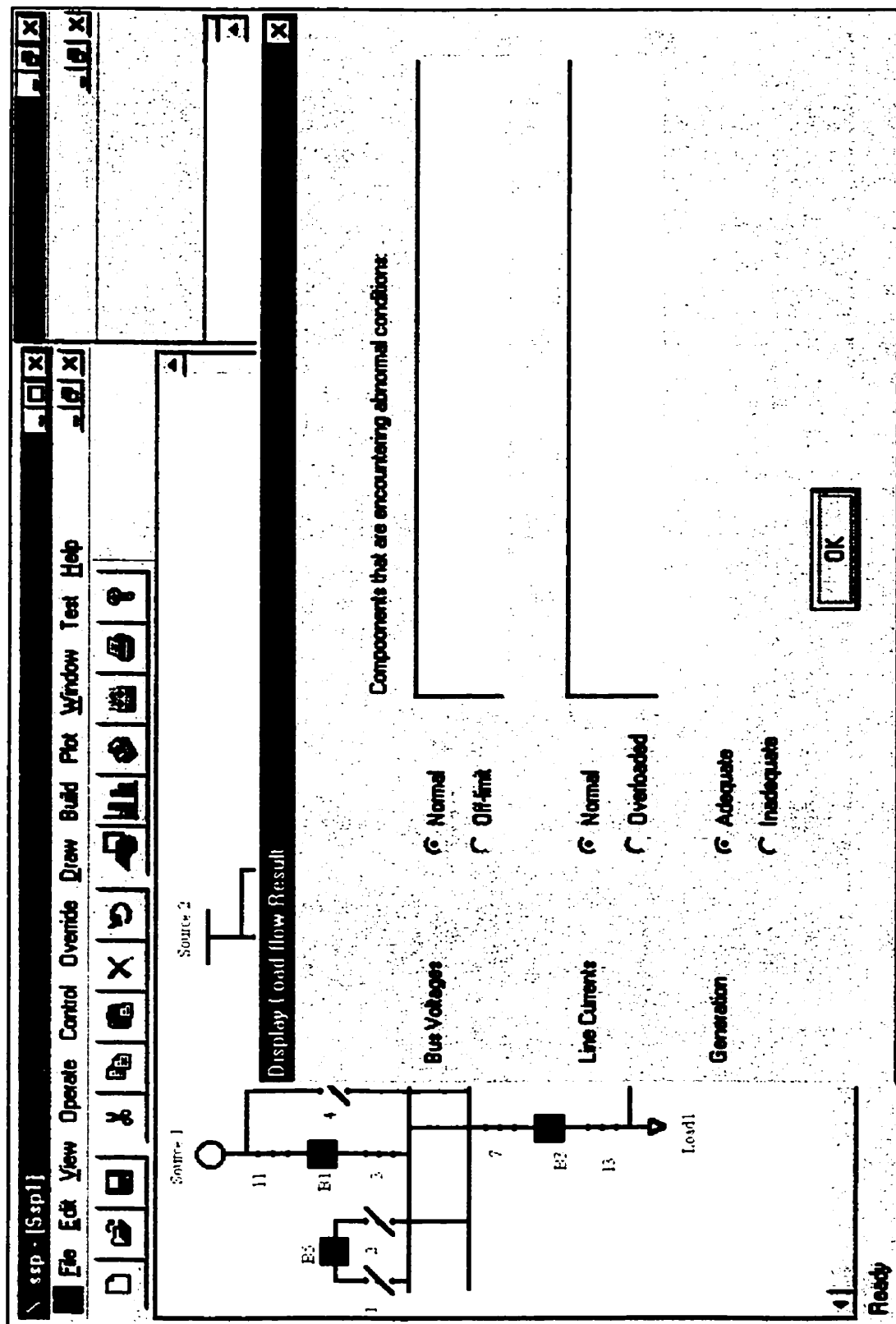


Figure H.76. Conclusion of the load-flow analysis.

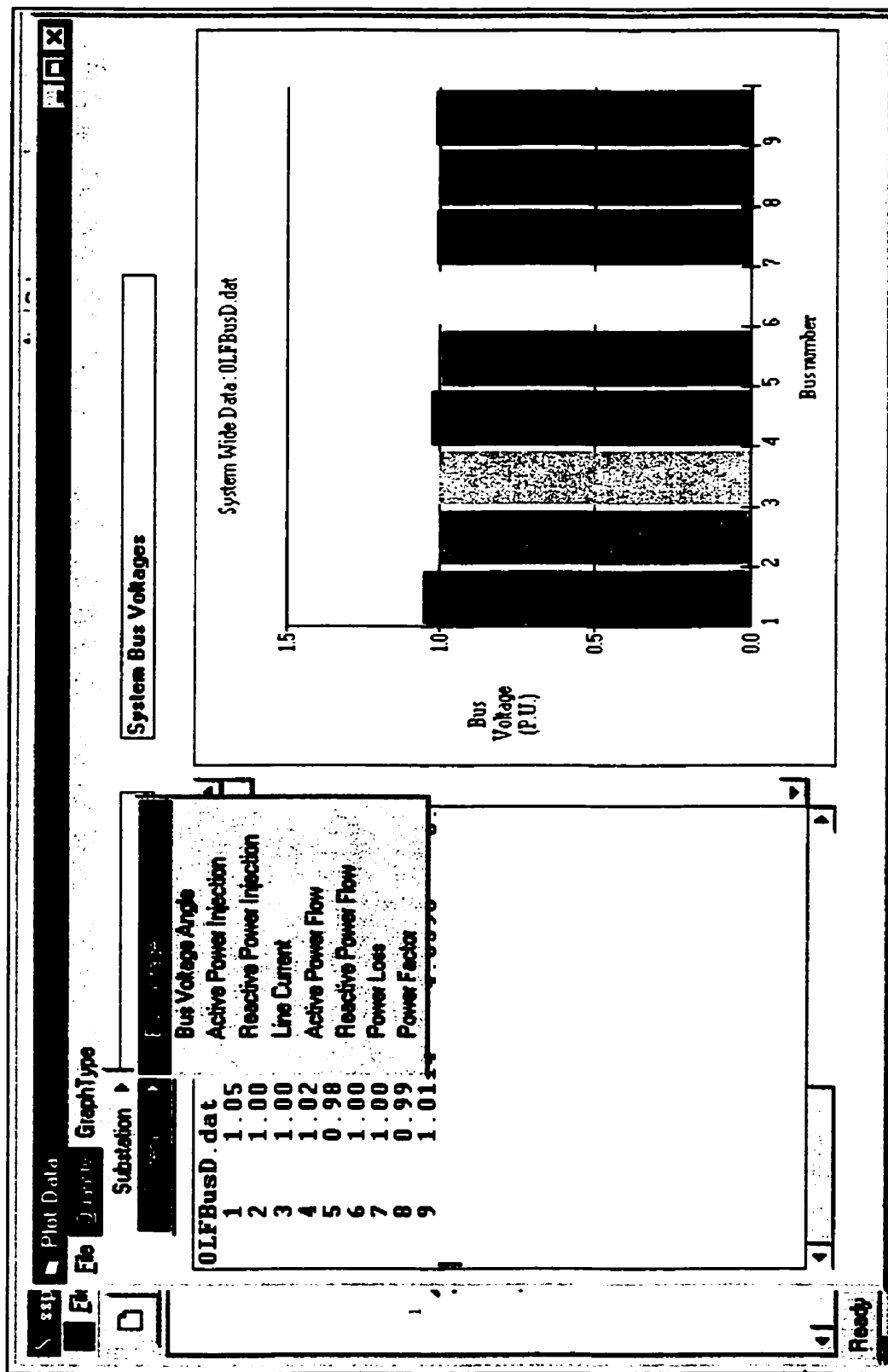


Figure H.77. Command being selected to plot various quantities and a plot of the voltage levels at system buses.

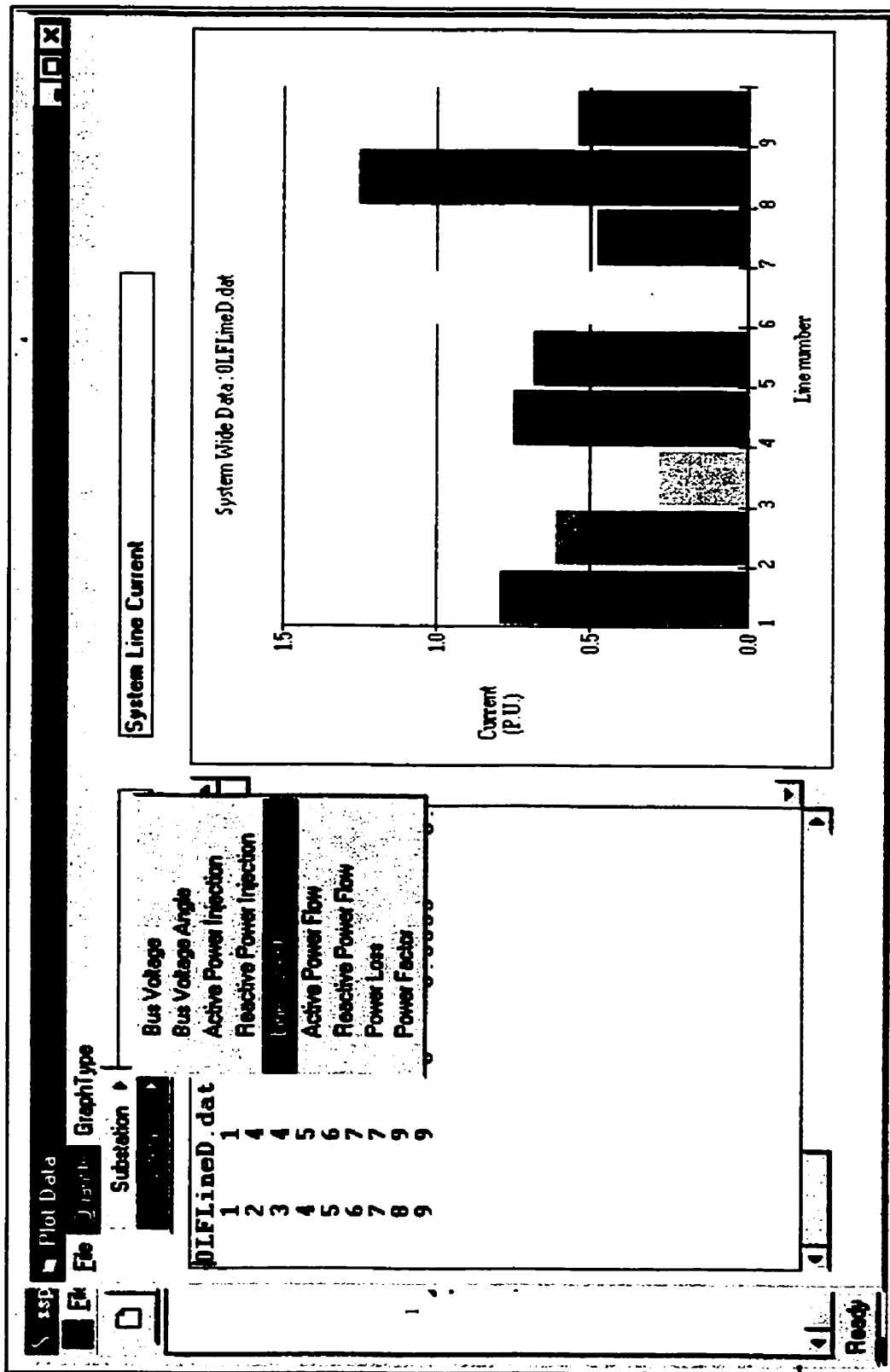


Figure H.78. A plot of the line current magnitudes.

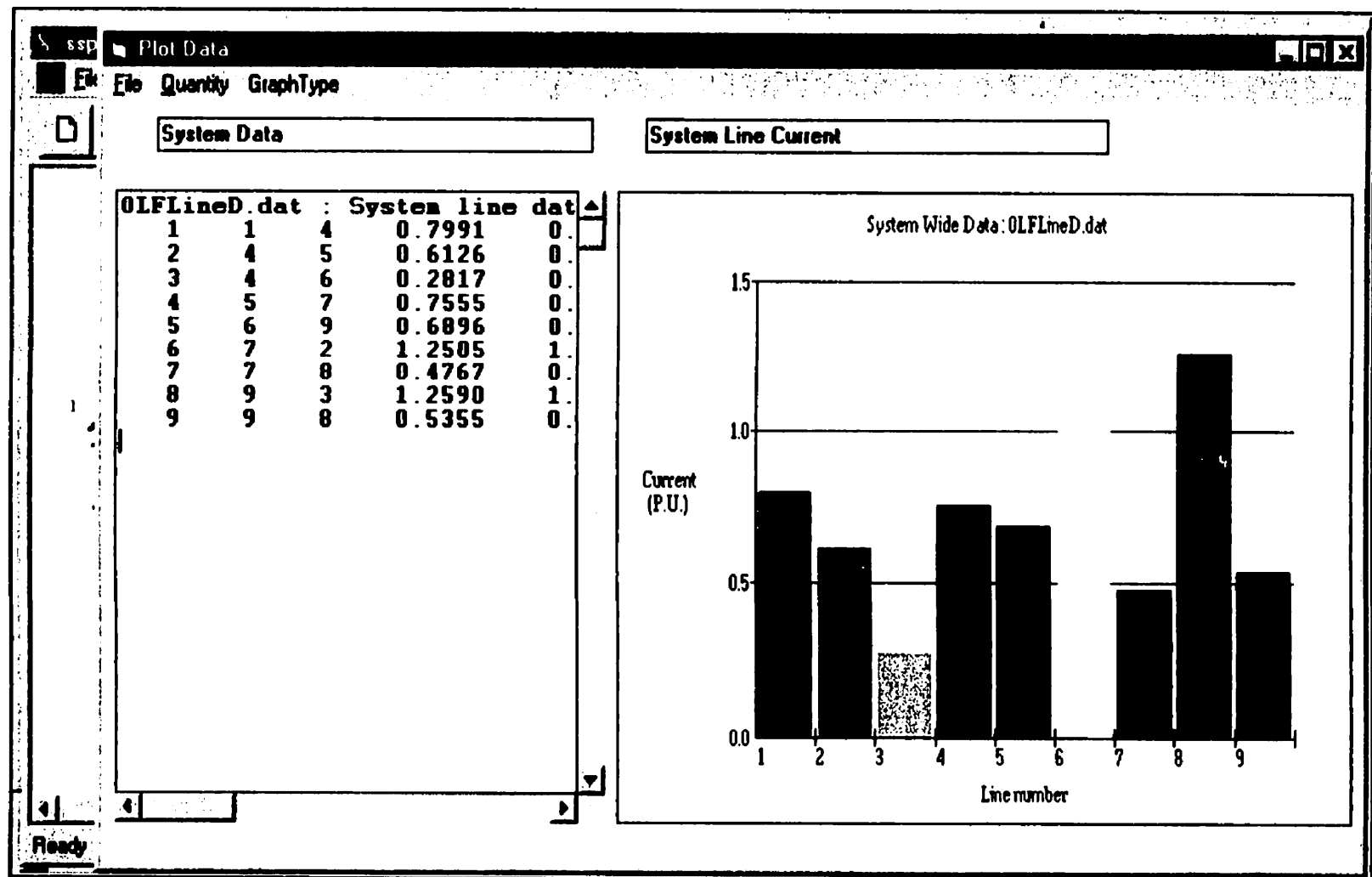


Figure H.79. Display of the load-flow output data in numeric form along with the graphical plot.